

УДК 504:002,004.45:[504.064(477):349.6(4-672 ЕС)]

КП 87.01.29

№ держреєстрації 0122U201790

Інв. №

МІНІСТЕРСТВО ЗАХИСТУ ДОВКІЛЛЯ ТА ПРИРОДНИХ РЕСУРСІВ УКРАЇНИ
НДУ "УКРАЇНСЬКИЙ НАУКОВИЙ ЦЕНТР ЕКОЛОГІЇ МОРЯ"
(УкрНЦЕМ)

65009, м. Одеса, Французький бульвар, 89. тел. (0482) 63 66 22, факс (0482) 637322
e-mail: accem@te.net.ua, www.sea.gov.ua



ЗВІТ

ПРО НАУКОВО-ДОСЛІДНУ РОБОТУ

**РОЗРОБКА ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ ВИКОНАННЯ
ЗАВДАНЬ МОРСЬКОЇ СТРАТЕГІЇ УКРАЇНИ У 2022 РОЦІ**

Науковий керівник НДР

Виконуючий обов'язки директора

УкрНЦЕМ, заступник директора з
науки, канд. геогр. наук, с. н. с.

Віктор КОМОРІН

2022

Рукопис закінчено 26 грудня 2022 р.

Результати роботи розглянуто Вченою Радою УкрНЦЕМ,
протокол від 19 січня 2023 № 1

СПИСОК АВТОРІВ

Науковий керівник НДР
 Виконуючий обов'язки директора
 Українського наукового центру
 екології моря, заступник директора з
 наукової роботи,
 канд. геогр. наук

В. М. Коморін
 (вступ;
 висновки;)


 " 26 " 12 2022

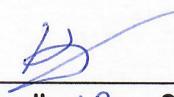
Начальник відділу
 управління екологічними даними

О. В. М'яснікова
 (розділ 5)


 " 26 " 12 2022

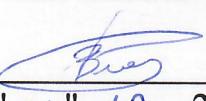
Завідувач сектором
 розробки інформаційних систем
 відділу інформаційних систем

А. М. Круглов
 (розділи 1, 2)


 " 26 " 12 2022

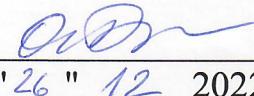
В.о. начальника відділу
 інформаційних систем

О. В. Леп'ошкін
 (розділи 1, 3, 4)


 " 26 " 12 2022

Науковий співробітник
 сектору геоінформаційного аналізу
 відділу інформаційних систем

О. С. Братченко
 (розділ 3)


 " 26 " 12 2022

РЕФЕРАТ

Звіт з НДР: 101 с., 9 рис., 7 джерел.

БАЗА ДАНИХ, ВЕБ-САЙТ, МОРСЬКА СТРАТЕГІЯ, ЕКОЛОГО-ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ, ІНТЕРАКТИВНЕ КАРТОГРАФУВАННЯ ІНТЕГРАЛЬНИХ ОЦІНОК СТАНУ МОРСЬКИХ ЕКОСИСТЕМ.

Актуальність науково-дослідної роботи обумовлюється, вперш за все необхідністю створення інформаційного забезпечення державного екологічного моніторингу морів України в межах створення та реалізації Морської стратегії України.

Метою НДР є удосконалення інформаційного забезпечення системи морського екологічного моніторингу в межах реалізації Морської стратегії України.

Основними завданнями НДР є:

1. Забезпечення функціонування апаратно-програмного комплексу для виконання науково-практичних завдань;
2. Розробка інтерфейсів баз даних (каталоги, аналітичні інструменти тощо);
3. Оновлення та забезпечення функціонування інтерактивної картографічної системи для візуалізації та аналізу інтегральних оцінок стану морських екосистем відповідно до вимог Рамкової Директиви ЄС про морську стратегію;
4. Забезпечення функціонування інформаційного веб-сайту УкрНЦЕМ;
5. Поповнення Бази Даних УкрНЦЕМ даними за 2022 р.

Об'єкт дослідження – інформаційне забезпечення екологічного моніторингу стану морського середовища Чорного моря в межах Морської

стратегії України відповідно до вимог Директиви 2008/56/ЄС щодо морської стратегії.

ЗМІСТ	С.
ПЕРЕЛІК СКОРОЧЕНЬ	6
ВСТУП	7
1. ЗАБЕЗПЕЧЕННЯ ФУНКЦІОNUВАННЯ АПАРАТНО-ПРОГРАМНОГО КОМПЛЕКСУ ДЛЯ ВИКОНАННЯ НАУКОВО-ПРАКТИЧНИХ ЗАВДАНЬ	10
1.1 Встановлення та налаштування апаратного забезпечення на серверному майданчику УкрНЦЕМ (новий сервер)	10
1.2 Прокладання нової локальної мережі для більш надійного доступу до мережевих ресурсів співробітниками УкрНЦЕМ	11
2. РОЗРОБКА ІНТЕРФЕЙСІВ БАЗ ДАНИХ (КАТАЛОГИ, АНАЛІТИЧНІ ІНСТРУМЕНТИ ТОЩО)	14
3. ОНОВЛЕННЯ ТА ЗАБЕЗПЕЧЕННЯ ФУНКЦІОNUВАННЯ ІНТЕРАКТИВНОЇ КАРТОГРАФІЧНОЇ СИСТЕМИ ДЛЯ ВІЗУАЛІЗАЦІЇ ТА АНАЛІЗУ ІНТЕГРАЛЬНИХ ОЦІНОК СТАНУ МОРСЬКИХ ЕКОСИСТЕМ ВІДПОВІДНО ДО ВИМОГ РАМКОВОЇ ДИРЕКТИВИ ЄС ПРО МОРСЬКУ СТРАТЕГІЮ	18
4. ЗАБЕЗПЕЧЕННЯ ФУНКЦІОNUВАННЯ ІНФОРМАЦІЙНОГО ВЕБ-САЙТУ УКРНЦЕМ	21
4.1 Забезпечення безпеки оновленого веб-сайту	21
4.2 Удосконалення елементів дизайну та сторонніх модулів	22
4.3 Наповнення веб-сайту інформацією	23
5. ПОПОВНЕННЯ БАЗИ ДАНИХ УКРНЦЕМ ДАНИМИ ЗА 2022 Р.	24
ВИСНОВКИ	26
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ	27
ДОДАТОК А «Код аналітичних інструментів»	29
ДОДАТОК Б «Код інтерактивної системи аналізу»	58

ПЕРЕЛІК СКОРОЧЕНЬ

БД – база даних;

ВЕЗ – виключна морська економічна зона України;

ВРД - Водна Рамкова Директива;

ГІС – геоінформаційна система;

ДЕС – добрий екологічний стан

ДЕММ – державний екологічний морський моніторинг

МОК – міжурядова океанографічна комісія;

МООДІ – Міжнародний Обмін океанографічними Даними та інформацією;

НДР – науково-дослідна робота;

ПЗЧМ – північно-західна частина Чорного моря;

ПАВ – поліциклічні ароматичні вуглеводні;

ПХБ – поліхлорбіфеніли;

РДМС – Рамкова директива про морську стратегію;

ХОП – хлорорганічні пестициди ;

УкрНЦЕМ – Український науковий центр екології морів;

DCT – Data Collection Templates;

IODE – International Oceanographic Data and Information Exchange Program of IOC;

IOC – Intergovernmental Oceanographic Commission of UNESCO;

HEAT – HELCOM Eutrophication Assessment Tool;

HELCOM – Балтійська комісія з охорони морського навколишнього середовища;

MSFD – Рамкова Директива про морську стратегію;

ВСТУП

Україні, як морській державі, потрібно створити систему управління якістю морського середовища на сучасному міжнародному рівні. В ЄС є, перевірені багатолітнім досвідом, такі інструменти, як Рамкова директива про морську стратегію (РДМС) - Директива 2008/56/ЄС Європейського парламенту та Ради Європи «Про встановлення рамок діяльності Співтовариства у сфері екологічної політики щодо морського середовища» від 17 червня 2008 року та Водна Рамкова Директива (ВРД) - Директива 2000/60/ЄС Європейського парламенту та Ради «Про встановлення рамок діяльності Співтовариства в галузі водної політики» від 23 жовтня 2000 року.

Відповідно до Угоди про асоціацію між Україною та ЄС Мінекоенерго з метою імплементації Директиви ЄС з морської стратегії необхідно здійснити заходи для визначення базового екологічного стану та статусу екосистем Чорного та Азовського морів в межах виключної морської економічної зони України (ВЕЗ), визначити та затвердити критерії добrego екологічного стану (ДЕС) для екосистем Чорного та Азовського морів в межах терitorіальних вод та ВЕЗ, визначити природоохоронні цілі та індикатори, досягнення яких має забезпечити наближення екологічного стану та статусу екосистем Чорного та Азовського морів в межах терitorіальних вод України та ВЕЗ до ДЕС [1], [2]. Все це повинно увійти до Морської стратегії України.

Український науковий центр екології моря (УкрНЦЕМ), відповідно до ст. 11 РДМС [3], та на основі базової оцінки, здійсненої відповідно до частини 1 ст. 8, розробив програму екологічного моніторингу для постійної оцінки екологічного стану морських вод, базуючись на переліках характеристик, видів джерел та наслідків впливу, зазначених у Додатках III і V РДМС [4].

Програма державного екологічного моніторингу морів України, розроблялась з урахуванням орієнтирів розвитку України як морської держави і пов'язаного з цим процесу інтеграції до ЄС, що потребує поступового

впровадження загальноєвропейських стандартів і зокрема директив у сфері водної політики.

Програма державного екологічного моніторингу створена відповідно до Порядку здійснення державного моніторингу вод, який затверджено Постановою Кабінету Міністрів України від 19 вересня 2018 р. № 758 [5] на виконання вимог РДМС та МРД, які Україна зобов'язалася імплементувати в межах виконання Угоди про асоціацію між Україною та ЄС.

Актуальність науково-дослідної роботи (НДР) обумовлюється, перш за все необхідністю створення інформаційного забезпечення державного екологічного моніторингу морів України в межах створення та реалізації Морської стратегії України.

Метою НДР є удосконалення інформаційного забезпечення системи морського екологічного моніторингу в межах реалізації Морської стратегії України.

Основні завдання НДР:

- забезпечення функціонування апаратно-програмного комплексу для виконання науково-практичних завдань;
- розробка інтерфейсів баз даних;
- оновлення та забезпечення функціонування інтерактивної картографічної системи для візуалізації та аналізу інтегральних оцінок стану морських екосистем відповідно до вимог рамкової директиви ЄС про морську стратегію;
- забезпечення функціонування інформаційного веб-сайту УкрНЦЕМ;
- поповнення бази даних УкрНЦЕМ даними за 2022 р.

Наукова робота здійснюється на базі попередніх розробок УкрНЦЕМ в межах національних та міжнародних програм. Дані, які були отримані протягом попередніх років є функціональною основою для наступних розробок та модифікацій.

Робота буде використовуватися в інформаційній діяльності Морського інформаційно-аналітичного центру УкрНЦЕМ для забезпечення державних організацій щодо прийняття оперативних і стратегічних управлінських рішень у сфері охорони морського середовища і прибережної смуги України та регулювання морського і прибережного природокористування, імплементації РДМС до основних напрямків досліджень, а також для забезпечення потреб широкого кола наукових співробітників і громадськості в екологічній інформації. Термін виконання НДР: 2022-2024 рр.

1. ЗАБЕЗПЕЧЕННЯ ФУНКЦІОNUВАННЯ АПАРАТНО-ПРОГРАМНОГО КОМПЛЕКСУ ДЛЯ ВИКОНАННЯ НАУКОВО-ПРАКТИЧНИХ ЗАВДАНЬ

Для забезпечення функціонування апаратного програмного комплексу УкрНЦЕМ було проведено наступний перелік робіт:

- 1) Встановлення та налаштування апаратного забезпечення на серверному майданчику УкрНЦЕМ (новий сервер);
- 2) Прокладання нової локальної мережі для більш надійного доступу до мережевих ресурсів співробітниками УкрНЦЕМ.

1.1 Встановлення та налаштування апаратного забезпечення на серверному майданчику УкрНЦЕМ (новий сервер)

Після встановлення нового апаратного забезпечення на серверному майданчику УкрНЦЕМ (новий сервер має потужні характеристики завдяки 16 процесору AMD Ryzen Threadripper PRO 3955WX із шістнадцятьма ядрами та 128 гігабайтами оперативної пам'яті, RAID-1 із двох твердотільних накопичувачів Samsung PM883 для операційної системи та RAID-10 із чотирьох жорстких дисків об'ємом 8TB) було прийнято рішення створити із нього сервер віртуалізації.

Для реалізації цього на сервер було встановлено гіпервізор VMware ESXi – програмний продукт для віртуалізації рівня підприємства.

ESXi є вбудованим гіпервізором і встановлюється безпосередньо на «голе залізо», тобто при установці не потребує наявності на машині встановленої операційної системи.

Гіпервізор ESXi дозволяє розділити ресурси фізичного комп'ютера на логічні розділи, які називаються віртуальними машинами. Включає засоби управління віртуальними машинами і ресурсами. Пред'являє певний набір вимог до апаратного забезпечення, зокрема, є обов'язковою наявність підтримки віртуалізації з боку процесора та материнської плати.

Для зберігання та розповсюдження віртуальних машин підтримує відкритий стандарт Open Virtualization Format.

Після цього, завдяки гіпервізору ESXi було створено декілька віртуальних машин під різноманітні потреби – веб-сервер(оновлення існуючого веб-серверу, та перенесення його конфігурації на новий сервер), сервер баз даних(оновлення існуючої СКБД Microsoft SQL Server та перенесення існуючої конфігурації на новий сервер), та декілька тестових віртуальних машин для тестування.

Завдяки функціоналу гіпервізора ESXi було отримано можливості швидко робити резервні копії віртуальних машин завдяки існуючим в УкрНЦЕМ інструментам створення резервних копій, розгорнати нові віртуальні машини як серверного класу, так і для простих користувачів, які зможуть працювати у режимі терміналу.

1.2 Прокладання нової локальної мережі для більш надійного доступу до мережевих ресурсів співробітниками УкрНЦЕМ.

Крім того було розгорнуто нову локальну мережу у будівлі УкрНЦЕМ. Нова локальна мережа розгорнута по топології зірка.

Топологія «зірка» — це єдина топологія мережі з явно виділеним центром, до якого підключаються всі інші абоненти. Обмін інформацією йде винятково через центральний концентратор, на який лягає більше навантаження, тому нічим іншим, крім мережі, він, як правило, займатися не

може. Зрозуміло, що мережне устаткування центрального абонента повинно бути істотно складнішим, чим устаткування периферійних абонентів. Проте рівноправність всіх абонентів (як у шині) у цьому випадку говорити не доводиться. Звичайно центральний комп'ютер найпотужніший, саме на нього покладають всі функції по керуванню обміном. Ніякі конфлікти в мережі з топологією зірка в принципі неможливі, тому що керування повністю централізоване.

Якщо говорити про стійкість зірки до відмов комп'ютерів, то вихід з ладу периферійного комп'ютера або його мережного встаткування ніяк не відбувається на функціонуванні мережі, зате будь-яка відмова центрального комп'ютера робить мережу повністю непрацездатною. У зв'язку із цим повинні прийматися спеціальні заходи щодо підвищення надійності центрального комп'ютера і його мережної апаратури.

Обрив кабелю або коротке замикання в ньому при топології зірка порушує обмін тільки з одним комп'ютером, а всі інші комп'ютери можуть нормальню продовжувати роботу.

На відміну від шини, у зірці на кожній лінії зв'язку перебувають тільки два абоненти: центральний й один з периферійних. Найчастіше для їхнього з'єднання використовується дві лінії зв'язку, кожна з яких передає інформацію в одному напрямку, тобто на кожній лінії зв'язку є тільки один приймач й один передавач. Це так звана передача точка-точка. Все це істотно спрощує мережне встаткування в порівнянні із шиною й рятує від необхідності застосування додаткових, зовнішніх термінаторів.

Велика перевага зірки полягає в тому, що всі точки підключення зібрані в одному місці. Це дозволяє легко контролювати роботу мережі, локалізувати несправності шляхом простого відключення від центра тих або інших абонентів (що неможливо, наприклад, у випадку шинної топології), а також обмежувати доступ сторонніх осіб до життєво важливих для мережі точок підключення. До периферійного абонента у випадку зірки може підходити як один кабель (по якому йде передача в обох напрямках), так і два (кожний

кабель передає в одному із двох зустрічних напрямків), причому останнє зустрічається набагато частіше.

Загальним недоліком для всіх топологій типу є значно більша, ніж при інших топологіях, витрата кабелю. Наприклад, якщо комп'ютери розташовані в одну лінію, то при виборі топології зірка знадобиться в кілька разів більше кабелю, ніж при топології шина. Це істотно впливає на вартість мережі в цілому й помітно ускладнює прокладку кабелю.

Нова локальна мережа УкрНЦЕМ побудована за вище зазначеною технологією за допомогою апаратного забезпечення Mikrotik. Головний керуючий пристрій – Mikrotik CCR1016-12S-1S+. Він керує всіма основними процесами у локальній мережі – надання адреси у локальній мережі кожному пристрою(завдяки DHCP-серверу), надання таблиці імен для доступу до ресурсів УкрНЦЕМ(DNS-сервер). Кожен поверх являє собою окремий «промінь» зірки, має свою власну адресацію, свій власний DHCP-сервер, які надаються завдяки Mikrotik CCR1016-12S-1S+. На кожному поверсі встановлено керований маршрутизатор Mikrotik RB2011UiAS-2HnD із можливістю надання Wi-Fi доступу до мережі. Саме завдяки керованому маршрутизатору Mikrotik RB2011UiAS-2HnD здійснюється підключення окремих кабінетів до локальної мережі.

Завдяки всьому цьому можна дуже легко виявляти будь-які негаразди у локальній мережі, фільтрувати трафік, обмежувати доступ або швидкість доступу до будь-яких ресурсів у будь-якому «проміні» незалежно від кожного іншого «променю», або й взагалі від'єднати «промінь» від локальної мережі не пошкодивши всю мережу взагалі.

2. РОЗРОБКА ІНТЕРФЕЙСІВ БАЗ ДАНИХ (КАТАЛОГИ, АНАЛІТИЧНІ ІНСТРУМЕНТИ ТОЩО)

Нові інструменти аналізу означають вбудовані інструменти в базу даних, які, відповідно до наданих алгоритмів, виконують повністю автоматичний перерахунок усіх значень, наявних у базі даних, і будуть виставлені відповідні позначки якості. Після цієї процедури параметри відображення інформації в наявних виборах будуть змінені для відображення отриманих позначок якості.

Нове програмне забезпечення для розрахунку та позначення позначок якості буде розроблено на мові програмування C# у середовищі розробки Microsoft Visual Studio.

Для відображення інформації в існуючому веб-інтерфейсі необхідно буде змінити всі існуючі SQL-запити до бази даних і змінити алгоритми відображення інформації з цих запитів користувачеві мовою програмування PHP. Насправді ці зміни є кардинальними і внутрішніми, і це буде повна зміна та переосмислення ядра веб-інтерфейсу.

За допомогою інструментів і додаткових бібліотек плагінів мови PHP буде реалізована можливість графічного відображення даних з бази даних у веб-інтерфейсі. Для цього буде реалізовано додатковий пункт меню та окремий вибір даних (рис. 2.1), результатом якого буде графік, зображений на зображенні (рис. 2.2). На зображенні показано результат вибору аміачного азоту (NH_4) на поверхневому шарі (0-1 м) на всіх станціях круїзу JBSS GE-UA 2019. Таким чином, дуже легко побачити аномалії або загальні зміни певних параметрів на всіх станціях на заданих глибинах відбору проб.

Параметри вибірки дозволяють здійснювати гнучкий відбір – наприклад, можна буде відібрати станції з певного круїзу, станції за водоймами та зонами, станції для певних інтервалів часу. Крім того, вибір глибини буде доступний або за певним шаром (наприклад, Thermocline Layer), або шляхом

безпосереднього введення значення глибини та додавання відхилення цієї глибини для більш повного покриття.

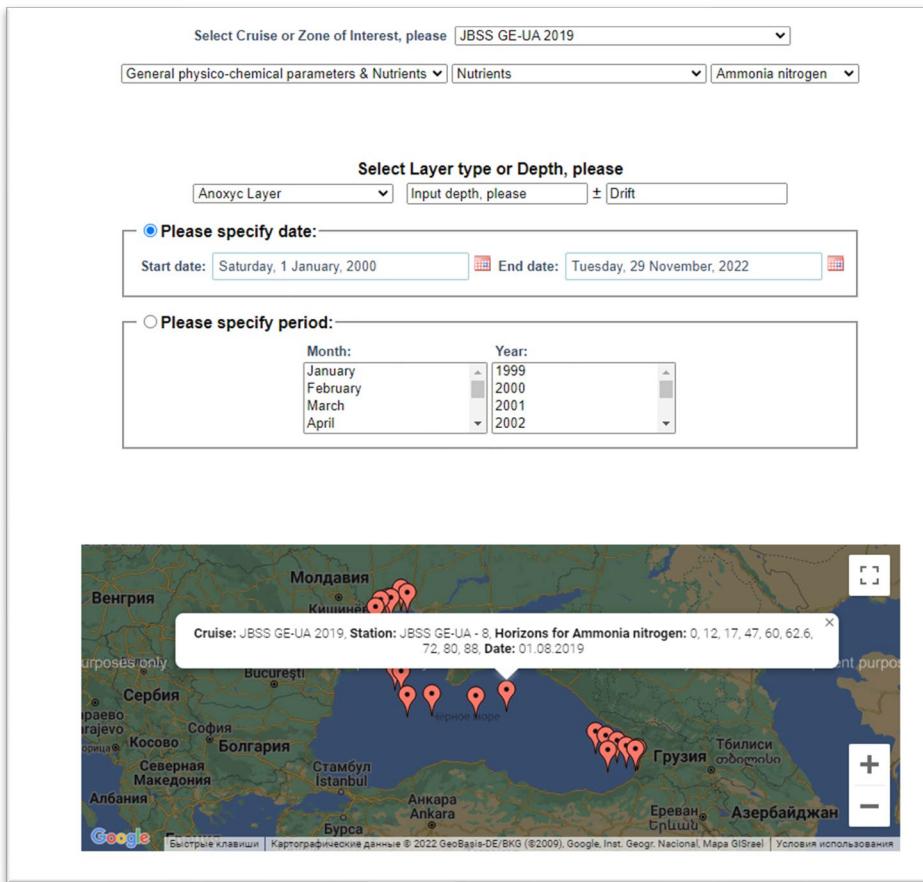


Рисунок 2.1 – Вибір інформації для побудови графіків.

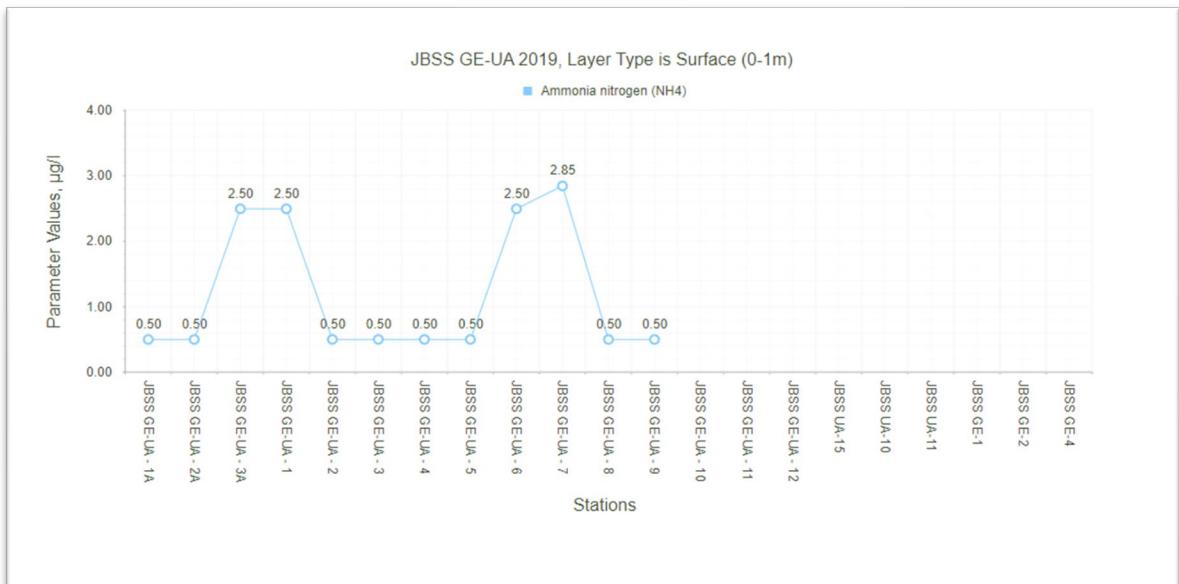


Рисунок 2.2 – Результатуючий графік.

На основі розроблених алгоритмів та їх реалізацій ми отримаємо досить гнучку систему оцінки даних у базі та веб-інтерфейсі на льоту. Таким чином, змінивши ядро виведення даних веб-інтерфейсу, можна буде виділяти дані сумнівної якості певними кольорами (наприклад, червоним або оранжевим), надійні дані (зеленим або синім) тощо.

В результаті відбору за заданими параметрами та в залежності від побажань користувача (виділяти значення з добору відповідно до пропорців якості чи ні) користувач отримує таблицю, в якій дані будуть підсвічуватися різними кольорів, що значно спростить аналіз і подальше використання даних, які зберігаються в базі даних.

Крім того було додано інструмент для обчислення стану морської води (GES) за допомогою Integral Zooplankton Index. Перш за все представлена можливість вибірки водних тіл та дат. Після встановлення необхідних параметрів на карті буде відображені вибрані станції та мета дані до них (рис 2.3). Подивитися метадані можна на кліку на станцію.

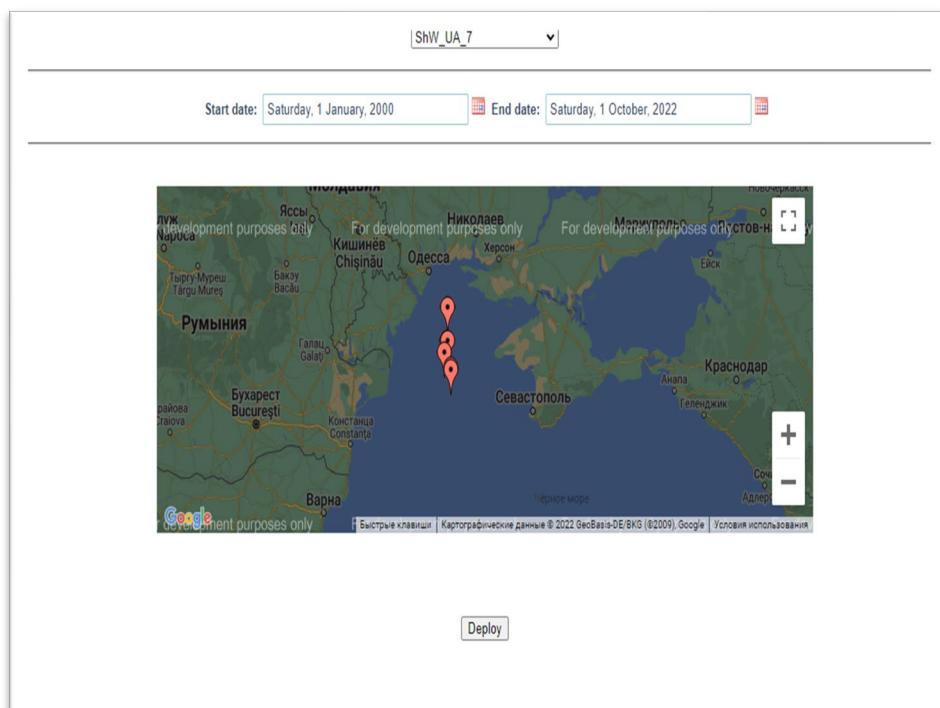


Рисунок 2.3 – Результат вибору.

За натисканням на кнопку відбувається перехід на нову сторінку, де представлені розрахунки по кожному горизонту станцій, що потрапили у вибірку. Ці розрахунки необхідні визначення Good Environmental Status (GES) у районі кожної станції, з наступним визначенням GES всього району загалом (рис. 2.4). Приклади кодів додано у Додатку А.

GES Not GES														
Meta Id	Station	Horizon	Total Biomass	Shannon	Copepoda	Noctiluka	Jelly	Total Biomass Ind	Shannon Ind	Copepoda Ind	Noctiluka Ind	Jelly Ind	IZI	
21	NPIMS UA - 9	0-12	34.651083333333	0.4862754737221	0.73141772864244	0.22962245047654	0.0298859126578534	0.87518394794940	0.99607807677776	0.73141772864244	0.77037754952346	0.99711408734215	0.99711408734215	
22	NPIMS UA - 9	12-30	25.652777777785	0.6805476856168	0.70735246345425	0	0.0001299404439618	0.87676400618096	0.1585053030	0.70735246345425	1	0.99967005955604	0.99967005955604	
23	NPIMS UA - 9	30-38	0.74575	2.6555495519245	0.93127723756019	0	0.69557499462311	0.758727843407	0.93127723756019	1	0.99967005955604	0.99967005955604		
74	JOSS GE-UA - 24-0-5	38 111+	2.7125294288660	0.72705775794794	0	0.000650599176887	0.26234836769950	0.87750093440605	0.77050093440605	0.69430720052111	0.73762633250503	0.99632260127759	0.99632260127759	
75	JOSS GE-UA - 24-0-5	38 111+	0.18655471610	0.266768247159	0.048456250529	0.048456250529	0.048456250529	0.048456250529	0.048456250529	0.048456250529	0.048456250529	0.99632260127759	0.99632260127759	
76	JOSS GE-UA - 24-25-60	123.6156974025	0.266768247159	0.7573148902729	0.048456250529	0.048456250529	0.048456250529	0.048456250529	0.048456250529	0.048456250529	0.048456250529	0.99632260127759	0.99632260127759	
77	JOSS GE-UA - 25-0-4	26 723	2.0317855707626	0.27440781345999	0.53542820375108	0.188002843599207	0.87654901757243	0.58051016300724	0.27440781345999	0.46405717920952	0.61599715600753	0.99632260127759	0.99632260127759	
78	JOSS GE-UA - 25-5-10	2 28325	2.549466732092	0.56941859191941	0.4157994670973	0.08958243277759	0.7842762091692	0.56941859191941	0.58420015329207	1	0.999715600753	0.999715600753		
79	JOSS GE-UA - 25 17-30	53.7253488372992	0.3693722040575	0.3742804704375	0.60585952852929	0.0082022113621246	0.87288381991753	0.67696348687358	0.3742804704375	0.3941407416706	0.99177768637368	0.99177768637368		
126	JOSS GE-UA - 2	0-17	83.605067494574	0.1101973341935	0.81249275911458	0	0.15532246001787	0.8705708256386	0.174420548539	0.81249275911458	1	0.84467753968213	0.84467753968213	
127	JOSS GE-UA - 2	17-34	207.345455043623	0.349968371018	0.89406387645745	0	0.01396078474472	0.07119981505309	0.65653941433165	0.95719953457472	0.89406387645745	0.98633921526553	0.98633921526553	
128	JOSS GE-UA - 2	34-51	0.18655471610	0.266768247159	0.048456250529	0	0.0155448303887665	0.067800781505309	0.65653941433165	0.95719953457472	0.89406387645745	0.98633921526553	0.98633921526553	
129	JOSS GE-UA - 3	0-20	147.600816788650	0.384118903068	0.897348811461	0	0.0239028166596	0.86780077216150	0.022654040087	0.86780077216150	0.98445516911233	0.98445516911233		
130	JOSS GE-UA - 3	20-35	29.853323808519	0.7772506441424	0.86473833212797	0	0.01584275789096	0.10565275637536	0.87566115223921	0.17921448978	0.86473833212797	0.886157242019	0.89013425325254	
131	JOSS GE-UA - 3	35-62	9.8236253288599	0.0832621724522	0.8299654117401	0	0.024736285420673	0.14378288764749	0.88182800717341	0.163663349863	0.82996641419079	0.975263714757932	0.9562171123521	
1196	JBSS GE-UA - 2	0-15	79.0211228982493	0.71109563712316	0.5981789954447	0.0868103560041296	0.1156997019192	0.87065544712155	0.062732499233	0.5981789954447	0.9913894399587	0.68430029608808	0.9913894399587	
1197	JBSS GE-UA - 2	15-35	68.065245077969	0.3538483572603	0.21014247584521	0.51141739018386	0.025059458367623	0.87164564838075	0.672528102436	0.21014247584521	0.48658260981614	0.97494054163238	0.9913894399587	
1198	JBSS GE-UA - 2	35-50	47.458764999671	0.67859177170106	0.16919014063998	0.0826551220713	0.0040579544919286	0.87353361137762	0.4795919343159	0.16919014063998	0.9173447879287	0.99594204550095	0.99594204550095	
1244	NMIS UA-17	0-15	51.66950273346	0.74704064495299	0.0319032307802650	0	0.88707700713614	0.55990912121513	0.2783075570084	0.0319032307802650	0.1129229968366	0.9933574430412	0.9933574430412	
1245	NMIS UA-17	15-27	22.6685710289475	0.3604102168021	0.75300683077055	0	0.09946255695878	0.87714143643	0.96011720480095	0.75300683077055	1	0.9933574430412	0.9933574430412	
1246	NMIS UA-17	27-40	507.72255051736	1.1636648139682	0.07836404975082	0.001380599769683	0.87100135495907	0.86119948229445	0.33247566113378	0.07836404975082	0.99661940042303	0.1289984530193	0.99661940042303	

Station Name	Integral Zooplankton Index
NPIMS UA - 9	0.999715600753
JOSS GE-UA - 24	0.999715600753
JOSS GE-UA - 2	0.999715600753
JOSS GE-UA - 3	0.999715600753
JOSS GE-UA - 3	0.999715600753
JBSS GE-UA - 2	0.999715600753
NMIS UA-17	0.999715600753

ShW_UA_7 (Open Water) Not GES 0.37943663701613

Рисунок 2.4 – Результат розрахунку GES за допомогою Integral Zooplankton Index.

3. ОНОВЛЕННЯ ТА ЗАБЕЗПЕЧЕННЯ ФУНКЦІОНАВАННЯ ІНТЕРАКТИВНОЇ КАРТОГРАФІЧНОЇ СИСТЕМИ ДЛЯ ВІЗУАЛІЗАЦІЇ ТА АНАЛІЗУ ІНТЕГРАЛЬНИХ ОЦІНОК СТАНУ МОРСЬКИХ ЕКОСИСТЕМ ВІДПОВІДНО ДО ВИМОГ РАМКОВОЇ ДИРЕКТИВИ ЄС ПРО МОРСЬКУ СТРАТЕГІЮ

Для збільшення функціоналу та стабільності роботи інтерактивної картографічної системи було здійснено оновлення картографічного проекту, що містить у собі просторові дані. Він був доопрацьований у зв'язку з оновленням самих просторових даних – оновлено символіку, дані уточнено та додано нові. Приклад виглядає інтерактивна картографічна система можна побачити на рисунку 3.1.

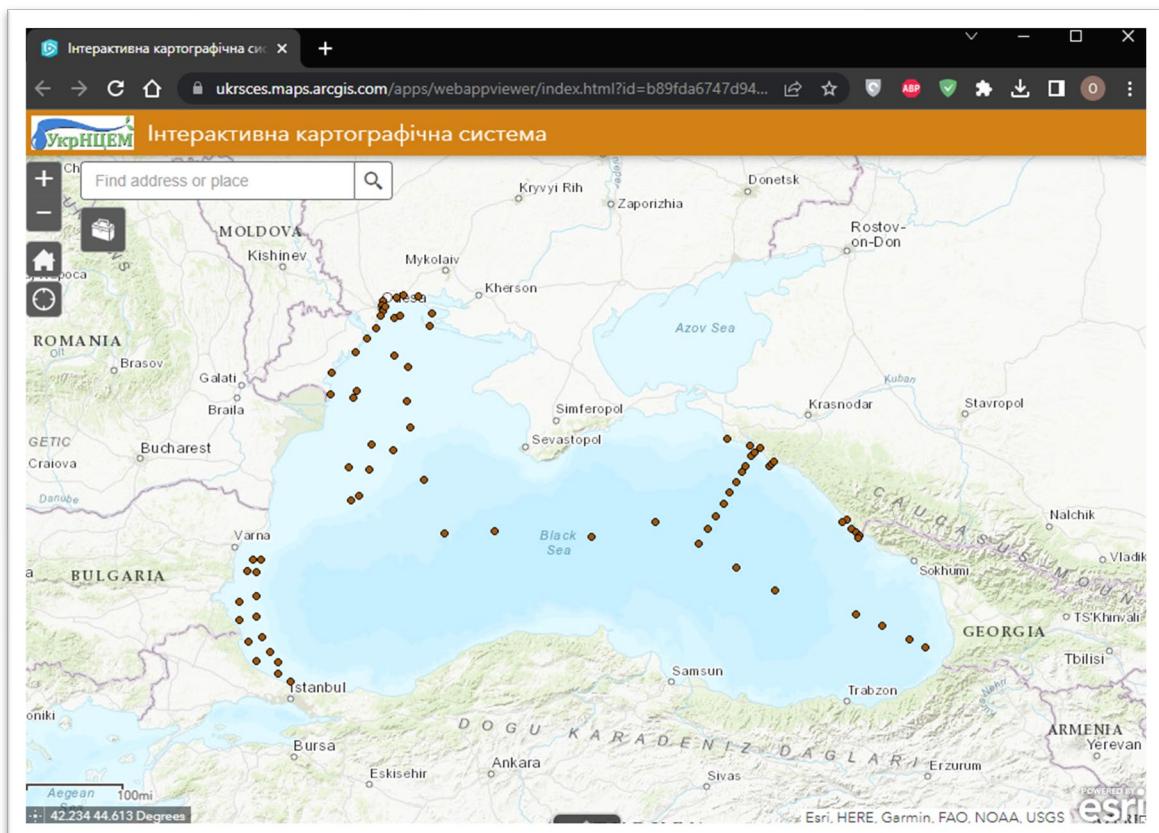


Рисунок 3.1 – Інтерактивна картографічна система

Також було оновлено до актуальної версії ESRI Web Application Builder, за допомогою якого відображається вся картографічна інформація онлайн та доступні інструменти для проведення різноманітного просторового аналізу.

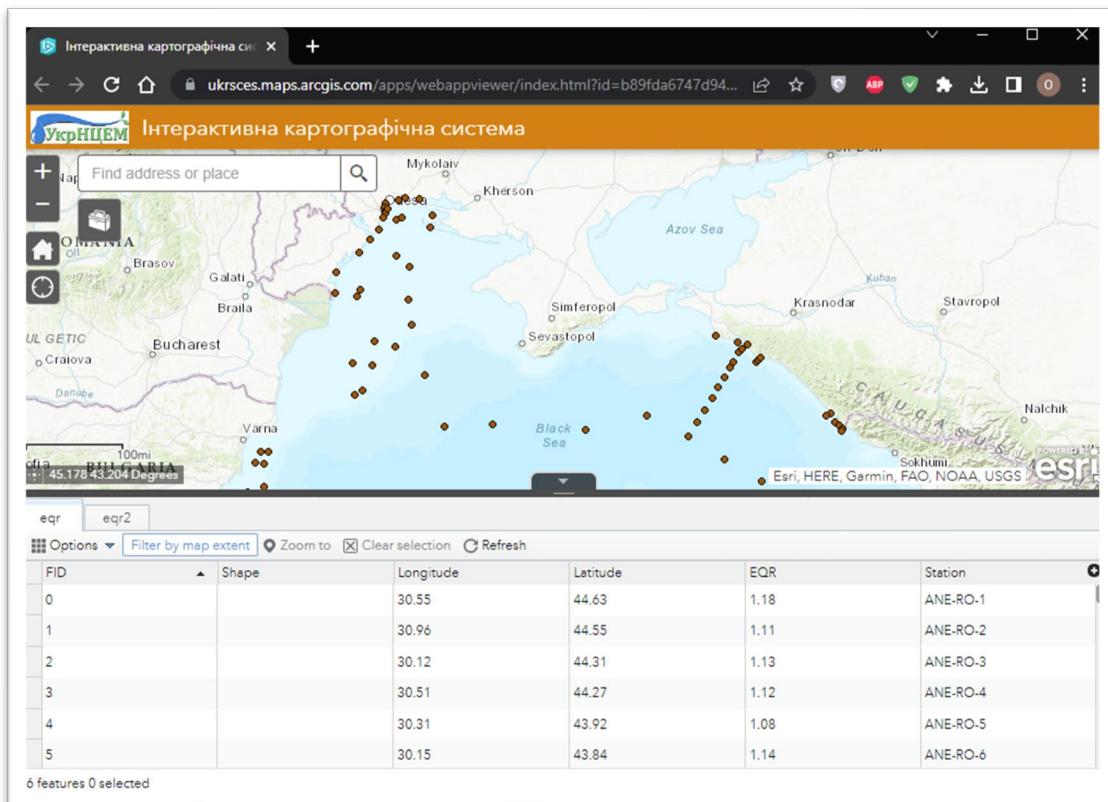


Рисунок 3.2 – Інструмент відображення атрибутивної інформації.

Дизайн інтерактивної картографічної системи зазнав деяких змін і став дружнішим у сприйнятті. Була додана можливість перегляду атрибутивної інформації кожного активного шару на інтерактивній карті, яка представлена на рисунку 3.2. Крім того, цей інструмент дозволяє фільтрувати інформацію, підсвічувуючи потрібну при виборі необхідних умов.

Так, наприклад, при заданні умови, якщо значення параметра EQR більше 2, колір символів не змінюється. Інші об'єкти, які не підпадають під цей фільтр змінюють колір на інший – у цьому випадку – зелений. Приклад завдання умови є на рисунку 3.3. Результат вибірки та роботи фільтра представлений рисунку 3.4. У Додатку Б представлено програмний код, який відповідає за новий функціонал..

Також у майбутньому буде додано можливість експорту даних, у тому числі й відсортованих.

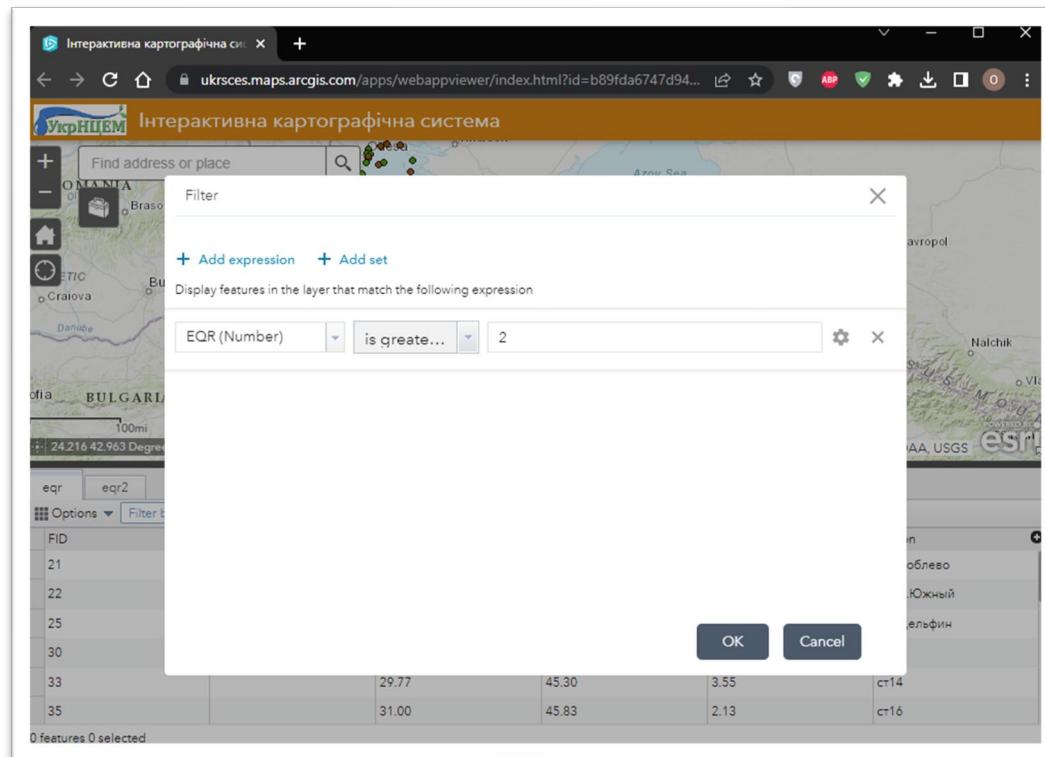


Рисунок 3.3 – Завдання умови на фільтр даних.

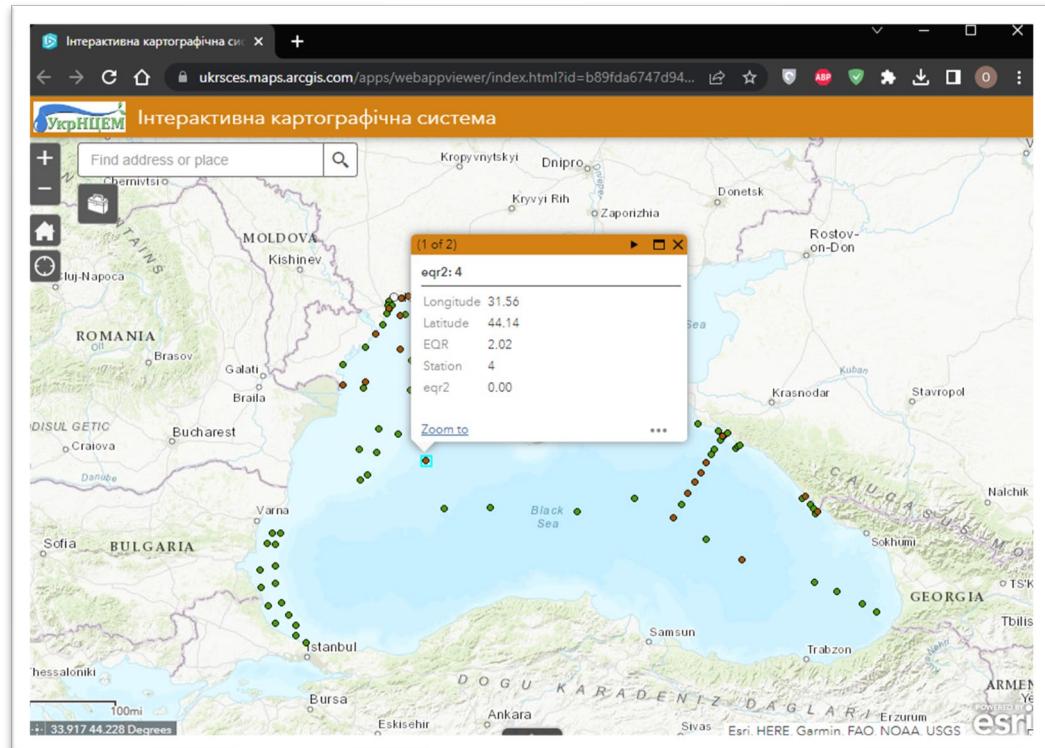


Рисунок 3.4 – Результат відфільтрованих даних.

4. ЗАБЕЗПЕЧЕННЯ ФУНКЦІОNUВАННЯ ІНФОРМАЦІЙНОГО ВЕБ-САЙТУ УКРНЦЕМ

Метою виконаної роботи є підтримка веб-сайту УкрНЦЕМ після його введення в експлуатацію, а також забезпечення інформаційної безпеки та зручності користування для кінцевого користувача. Новий веб-сайт має гідно представляти нашу організацію як на національному рівні, так і в міжнародній спільноті.

4.1. Забезпечення безпеки оновленого веб-сайту

Одним з пріоритетних напрямків під час розробки нового сайту, стало забезпечення інформаційної та технічної безпеки. Виконану впродовж року роботу в цьому напрямку можна поділити на наступні категорії:

- оновлення системи керування вмістом (CMS) WordPress останньої актуальної версії 5.3.2, шаблону Avada Design Tool версії 6.1, допоміжних модулів. Підтримка актуальних версій усіх модулів веб-сайту дозволяє забезпечити захист від програмних вразливостей, що виявляють самі розробники та користувачі. Ці дії необхідні, оскільки через програмні вразливості здійснюється більша частина усіх хакерських атак;
- забезпечення резервного копіювання файлів веб-сайту, бази даних та веб-сервісів. Резервне копіювання проходить в автоматичному режимі, усі дані копіюються на 2 різних сервери (основний та сервер резервних копій) 2-3 рази на тиждень, в залежності від типу даних. Базове резервне копіювання відбувається на основний сервер, що знаходиться під управлінням операційної системи (ОС) Microsoft Windows Server, після чого дублюється на спеціальний сервер резервних копій, на якому встановлено ОС на базі Linux». Крім того,

усі резервні копії зберігаються у декількох версіях, з різною датою створення. Таким чином, навіть якщо до резервної копії потраплять данні, що вже піддалися зараженню зловмисниками, все одно, зберігається більш стара, але не заражена копія даних. Ці заходи дозволяють відновити роботу системи у випадку її пошкодження або зламу;

– забезпечення програмного захисту на стороні серверу. Заходи, що включають у себе налаштування політики безпеки в залежності від користувача та його групи, закриття доступу до системних файлів із зовнішньої мережі – спрямовані на закриття можливих шляхів несанкціонованого доступу до веб-сайту (файлів та БД).

4.2 Удосконалення елементів дизайну та сторонніх модулів

За основу для побудови дизайну використовується набір модулів, шаблонів та інструментів під назвою «Avada Design Tool», що розробляється та підтримується компанією «Theme Fusion». Модулі, що входять до складу цього набору, дозволяють швидко та зручно організувати структуру та дизайн веб-сайту. Головна сторінка нового веб-сайту УкрНЦЕМ відображена на рисунку 4.1.

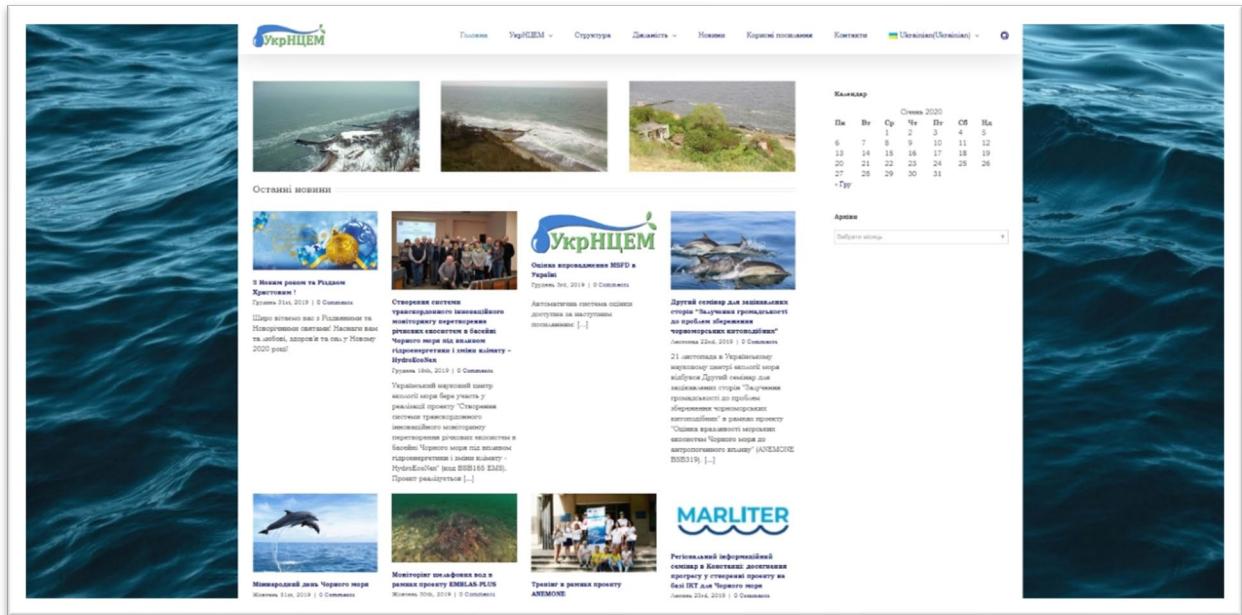


Рисунок 4.1 – Головна сторінка нового веб-сайту УкрНЦЕМ.

4.3 Наповнення веб-сайту інформацією

Після запуску веб-сайту для публічного доступу, проведено його наповнення новим або актуалізованим контентом, а саме:

- додано новини починаючи з початку 2019 року (16 новин);
- додано інформацію в розділ «Бази даних»;
- додано інформацію в розділ «ГІС»;
- додано інформацію в розділ «Проекти»;
- додано фотографії до галереї, що відображається на головній сторінці веб-сайту.

5. ПОПОВНЕННЯ БАЗИ ДАНИХ УКРНЦЕМ ДАНИМИ ЗА 2022 Р.

Метою роботи та основним завданням є розробка системи збору, поповнення, критконтролю і завантаження до БД «SeaBase» даних за програмою морського прибережного моніторингу та проектом «Контрольні спостереження в процесі експлуатації глибоководного суднового ходу Дунай-Чорне море (морська частина)»: «Гідрологічні та гідрохімічні спостереження за станом морського середовища в процесі експлуатації глибоководного суднового ходу Дунай-Чорне море».

Протягом звітного періоду було виконано наступні роботи:

- Аналіз отриманих УкрНЦЕМ даних за програмою моніторингу у 2021 році для подальшого внесення у БД «SeaBase»:
 - перевірка координат станцій та створення карт рейсів;
 - перевірка горизонтів та глибин станцій;
 - корегування одиниць вимірювання для забруднюючих речовин у воді, донних відкладах та біоті;
 - перевірка дат станцій моніторингу.
- Таблиця Platform оновлена двома рейсами за травень та жовтень 2021 року виконаного в рамках НДР Контрольні моніторингові спостереження в процесі експлуатації глибоководного суднового ходу Дунай-Чорне море (морська частина) спільно з Інститутом морської біології НАН України – «Дунай-Чорне море» з дванадцятьма станціями у кожному рейсі;
- Внесено 122 станції Прибережного моніторингу 2021 року у таблицю Stations бази даних УкрНЦЕМ «SeaBase».
- Внесенні дані екологічного моніторингу Чорного моря УкрНЦЕМ для 17 параметрів гідрології та гідрохімії за 2021 р. в таблицю Samples БД «SeaBase»:
 - 40 станцій «Пляж "Аркадия"»;

- 40 станцій «мис Малий Фонтан»;
 - 12 станцій Дунай травень;
 - 12 станцій Дунай жовтень;
 - 18 станцій прибережного моніторингу.
- Внесенні дані екологічного моніторингу Чорного моря УкрНЦЕМ для 77 параметрів забруднюючих речовин у воді за 2021 р. в таблицю Samples БД «SeaBase»:
 - 2 станції «Пляж "Аркадия"»;
 - 2 станції «мис Малий Фонтан»;
 - 12 станцій Дунай травень;
 - 12 станцій Дунай жовтень;
 - 18 станцій прибережного моніторингу.
 - Внесенні дані екологічного моніторингу Чорного моря УкрНЦЕМ для 82 параметрів забруднюючих речовин у донних відкладеннях за 2021 р. в таблицю Samples БД «SeaBase»:
 - 12 станцій Дунай травень;
 - 12 станцій Дунай жовтень.
 - Внесенні дані для п'яти параметрів фотосинтетичних пігментів у таблицю Samples БД «SeaBase» за 2021 рік:
 - 38 станцій «Пляж "Аркадия"»;
 - 37 станцій «мис Малий Фонтан»;
 - 16 станцій прибережного моніторингу.

ВИСНОВКИ

В ході НДР було виконано основний комплекс дій, які включали у себе повне оновлення локальної мережі та її апаратного забезпечення. Локальна мережа УкрНЦЕМ отримала ще більше протоколів захисту, які захищатимуть дані від несанкціонованого доступу ззовні та зсередини. Крім того було виконано основні вимоги до налаштувань безпеки даних, такі як конфіденційність, цілісність, доступність, достовірність, відмовостійкість на апаратному рівні виконано у повному об'ємі. Всі веб-ресурси УкрНЦЕМ зараз захищено асинхронним шифруванням.

Було розроблено інструменти для автоматичного аналізу даних, що знаходяться у базі даних УкрНЦЕМ. Завдяки цим автоматичним інструментам стало можливим в динаміці проводити необхідні розрахунки для визначення Good Environmental Status (GES) у районі кожної станції, з наступним визначенням GES всього району загалом.

У звітному періоді було оновлено картографічну систему з робочою назвою «Інтегральні оцінки стану морських екосистем». Додано новий функціонал, який надає можливість оперувати просторовими даними на новому рівні – робити вибірки, експортувати дані тощо. Також було перероблено інтерфейс картографічної системи, були придбані всі недоліки роботи системи.

На протязі звітного періоду на постійній основі оновлювався сайт УкрНЦЕМ та його функціональні можливості й продуктивність. Стрічка новин сайту оновлюється на постійній основі.

Здійснено поповнення, критконтроль і завантаження до БД «SeaBase» даних за програмою морського прибережного моніторингу. Оброблено и завантажено в базу наступні данні за 2021 рік: Гідрологія та гідрохімія -2268 записів; забруднюючі речовини у воді – 4596 записів; забруднюючі речовини (дно) – 1756 записів; фотосинтетичні пігменти - 445 записів. Загальна кількість записів - 9065.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Звіт про науково-дослідну роботу «Базова оцінка та визначення Доброго екологічного стану (ДЕС) морського середовища Чорного моря в межах виключної морської економічної зони України» у 6-ті томах [Текст] / Науковий керівник В.В. Український // Одеса, УкрНЦЕМ, 2018. – 636 с. Державна реєстрація № 0118U006641.
2. Звіт про науково-дослідну роботу «Базова оцінка та визначення Доброго екологічного стану біоценозів і біорізноманіття Чорного моря в межах виключної морської економічної зони України» [Текст] / Науковий керівник С.П. Ковалишина // Одеса, УкрНЦЕМ, 2018. – 138 с. Державна реєстрація № 0118U006642.
3. Directive 2008/56/EC of the European Parliament and of the Council of 17 June 2008, establishing a framework for community action in the field of marine environmental policy (Marine Strategy Framework Directive) [Text] // Official Journal of the European Union, 25.6.2008. P. 19-40.
4. Звіт про науково-дослідну роботу «Розроблення Програми державного екологічного моніторингу морів України на 2019-2025 рр. відповідно до вимог Директив ЄС 2008/56/ЄС, 2008/105/ЄС» [Текст] / Науковий керівник В. М. Коморін// Одеса, УкрНЦЕМ, 2019. – 387 с. Державна реєстрація №0118U006644.
5. Кабінет Міністрів України, Постанова від 19 вересня 2018 р. № 758, Київ «Про затвердження Порядку державного моніторингу вод» [Електронний ресурс]. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/758-2018-п> – Назва з екрану.
6. European Commission, Official Website. [Електронний ресурс] : [https://es.europa.eu]. – : Електронні дані. – Режим доступу : https://ec.europa.eu/environment/marine/eu-coast-and-marine-policy/marine-strategy-framework-directive/index_en.htm

7. Звіт про науково-дослідну роботу «Розробка еколого-інформаційного забезпечення морської стратегії України у 2018 р. згідно Директиви ЄС про встановлення рамок діяльності Співтовариства у сфері екологічної політики щодо морського середовища» [Текст]/ Науковий керівник В.М. Коморін // Одеса: УкрНІЦЕМ, 2018. – 82 с. Державна реєстрація № 0118U006648.

Додаток А

«Код аналітичних інструментів»

```

<?php
    session_start();
    require_once ($_SERVER['DOCUMENT_ROOT']."/connectionEmblas.php");
    require_once ("selectedpanel/select_content.php");
    $params = array();
    $options = array( "Scrollable" => SQLSRV_CURSOR_KEYSET );

    include ("header.php");
?>
<!DOCTYPE html>
<html>
<head>
<title>Graph statistics</title>
<script
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyBPlZQE7xdR6fpWrVgVsc81a5546JTE6S8"
async defer></script>
<script type="text/javascript" src="map.js"></script>
<script type="text/javascript" src="func.js"></script>
<!--
<!--<link rel="stylesheet" href="/css/select.css">-->
</head>
<body>
<form id="form1" action="graph.php" method="post" target ="blank">
<center>
What do you prefer - cruises or zones:
<input type='radio' id='station' name='zone_stations' value='station'><label
for='period'><b>Cruises</b></label>
<input type='radio' id='zone' name='zone_stations' value='zone'><label
for='period'><b>Zones</b></label>
<br><br><br>
<label id = 'label_for_location' name = 'label_for_location' class='labelHorizontal' hidden>Select Cruise
or Zone of Interest, please</label>
<select id='location' name ='location' hidden> <option value='none' selected disabled hidden>Select a
One</option> </select>
<br><br>
<select id='matrix' name = 'matrix'>
<option value='none' selected disabled hidden>Select an Option</option>
<option value = '1'> General physico-chemical parameters & Nutrients</option>
<option value = '2'> Pollution in water</option>
<option value = '3'> Pollution in sediment</option>
<option value = '4'> Pollution in biota</option>
</select>
<select id="par_group" name="par_group"><option value='none' selected disabled hidden>Select a
Matrix first</option></select>
<select id="par" name="par"><option value='none' selected disabled hidden>Select a Parametr Group
first</option></select> <br><br><br><br>
<table id = 'dep' name= 'dep' hidden>
<tr>

```

```

<th colspan = 3 >Select Layer type or Depth, please</th>
</tr>
<tr>
    <td>
        <select id='layer_type' name ='layer_type' hidden><option value='none' selected
disabled hidden></option> </select>
    </td>
    <td>
        &nbsp
    </td>
    <td>
        <input type="text" name="depth" value = "Input depth, please"/>
    &#177;
        <input type="text" name="different" value = "Drift"/>
    </td>
</tr>

</table>

<div id="test" name="test"></div>

<?php
echo"

<table border = 0>
<tr>
<td> <fieldset>

                <legend><input type='radio' id='date'
name='date_period' value='date'><label for='date'><b>Please specify date:</b></label></legend>
                <label for='alternateDP_from' class='labelHorizontal'>
                    Start date:
                </label>
                <input type='text' id='alternateDP_from' size='30'
disabled='disabled' value='Tuesday, 1 January, 2019' class='ui-widget-content ui-corner-all'>
                    <input name='datebegin' type='hidden'
value="'.date("01.01.2019")."' id='datebegin'/>

                <label for='alternateDP_to' class='labelHorizontal'>
                    End date:
                </label>
                <input type='text' id='alternateDP_to' size='30'
disabled='disabled' value="'.date("l, j F, Y")."' class='ui-widget-content ui-corner-all'>
                    <input name='dateend' type='hidden'
value="'.date("d.m.Y")."' id='dateend'/><br/>
            </fieldset></td>
        </tr>
        <tr>
            <td>
                <fieldset>
                    <legend><input type='radio' id='period' name='date_period' value='period'><label
for='period'><b>Please specify period:</b></label></legend>

                <center>

```

```

<table border = 0>

<tr>
<td>
    <label for='month' class='labelVertical'>
        Month:
    </label>
    <select id='month' name='month[]'>
        <option value='1'>January</option>
        <option value='2'>February</option>
        <option value='3'>March</option>
        <option value='4'>April</option>
        <option value='5'>May</option>
        <option value='6'>June</option>
        <option value='7'>July</option>
        <option value='8'>August</option>
        <option value='9'>September</option>
        <option value='10'>October</option>
        <option value='11'>November</option>
        <option value='12'>December</option>
    </select>
<td>
    <label for='year' class='labelVertical'>
        Year:
    </label>
    <select id='year' name='year[]' class='select_content' multiple></select>
</td>

</tr>
</table>
</center>

</fieldset>
</td>
</tr>

</table>

<br><br><br><br><br> "
?>

<center>
    <div id="dvMap" style="width: 800px; height: 300px" />
</center>

</br></br></br></br>

<input type = "submit" name = "submit" value = "Deploy">

</form>

```

```

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-MrcW6ZMFYlzcLA8NI+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>
</body>

</html>

<?php
    session_start();
    require_once ($_SERVER['DOCUMENT_ROOT']."/connectionEmblas.php");
    require_once ("selectedpanel/select_content.php");
    $params = array();
    $options = array( "Scrollable" => SQLSRV_CURSOR_KEYSET );

    include ("header.php");

?>

<html>
<head>
<title>Integral Zooplankton Index</title>
<script
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyBPlZQE7xdR6fpWrVgVsc81a5546JTE6S8"
async defer></script>
<script type="text/javascript" src="map.js"></script>

<script type="text/javascript" src="func.js">
    var today = new Date();
    var curYear = today.getFullYear();
</script>
</head>

<body>
<form id="form1" action="izi_calc.php" method="post" target ="blank">

<?php
echo "<center><select id='zone' name='zone' class='select_content'>";
$sql_zones = "select idZone, zone from [stationsZone]";
$zones = sqlsrv_query($conn,$sql_zones,$params,$options);
while($row = sqlsrv_fetch_array($zones, SQLSRV_FETCH_ASSOC))
{
    echo '<option value="" . $row['idZone'] . ">' . $row['zone']. '</option>';
}
echo "</select><center>      ";
echo "
<fieldset>
    <legend><label for='date'><b>Please specify
date:</b></label></legend>
    <label for='alternateDP_from' class='labelHorizontal'>

```

```

        Start date:
        </label>
        <input type='text' id='alternateDP_from' size='30'
disabled='disabled' value='Tuesday, 1 January, 2019' class='ui-widget-content ui-corner-all'>
        <input name='datebegin1' type='hidden'
value="'.date("01.01.2019")."' id='datebegin1'/>

        <label for='alternateDP_to' class='labelHorizontal'>
        End date:
        </label>
        <input type='text' id='alternateDP_to' size='30'
disabled='disabled' value="'.date("l, j F, Y")."' class='ui-widget-content ui-corner-all'>
        <input name='dateend1' type='hidden'
value="'.date("d.m.Y")."' id='dateend1'/><br/>
</fieldset>
?>
<br><br>

<center>
    <div id="dvMap" style="width: 800px; height: 300px" />
</center>

</br></br></br></br>
<input type = "submit" name = "submit" value = "Deploy">

</form>
</body>

</html>

<?php
    session_start();
    require_once ($_SERVER['DOCUMENT_ROOT']."/connectionEmblas.php");
    require_once ("selectedpanel/select_content.php");
    $params = array();
    $options = array( "Scrollable" => SQLSRV_CURSOR_KEYSET );

    include ("header.php");

    $zone_type = 0;
    $zone_name = "";

    $id_zone = $_POST['zone'];
    $date_begin = $_POST['datebegin1'];
    $date_end = $_POST['dateend1'];

    $id_meta_arr = array();
    $st_name_arr = array();
    $horisons_arr = array();

    $total_bio_ind = array();

```

```

$total_shannon_ind = array();
$total_copepoda_ind = array();
$total_noctiluca_ind = array();
$total_jelly_ind = array();
$izi_calc = array();
$izi_stations = array();
$id_stations = array();
$id_station_names = array();

$sql = "select zone,[zoneType] from [stationsZone] where idZone in ($id_zone)";

$zone_name = sqlsrv_query($conn,$sql,$params,$options);
while($row = sqlsrv_fetch_array($zone_name, SQLSRV_FETCH_ASSOC)){
    $zone_type = $row['zoneType'];
    $zone_name = $row['zone'];
}

echo "<center>";
echo "<table border = '1'>
<tr>
    <th>Meta_Id</th>
    <th>Station</th>
    <th>Horizon</th>
    <th>Total Biomass</th>
    <th>Shennon</th>
    <th>Copepoda</th>
    <th>Noctiluka</th>
    <th>Jelly</th>
    <th>Total Biomass Ind</th>
    <th>Shennon Ind</th>
    <th>Copepoda Ind</th>
    <th>Noctiluka Ind</th>
    <th>Jelly Ind</th>
    <th>IZI</th>
</tr>";
$begin = explode(".", $date_begin);
$begin_for_sql = $begin[2].".$begin[1].".$begin[0]; //echo $begin_for_sql;echo "<br>";
$end = explode(".", $date_end);
$end_for_sql = $end[2].".$end[1].".$end[0];
$sql_date = " and (CONVERT (date,(Convert (varchar,Stations.DayStart) +'.'+Convert
(varchar,Stations.MonthStart) +'.'+Convert (varchar,Stations.YearStart)), 104) between '$begin_for_sql'
and '$end_for_sql')"; //echo($sql_date);

echo "</br></br><center>Zone name is <b>$zone_name</b> Dates between <b>". $begin[0] .".".
$begin[1] . ". ". $begin[2] . "</b> and <b>". $end[0] .". ". $end[1] . ". ". $end[2] . "</b> </center>
</br></br>";

if ($izi_st[$i] < 0.542 && $izi_st[$i] > 0.264) {$color = "#FFFFCC";} // moderate
if ($izi_st[$i] < 0.263 && $izi_st[$i] > 0.141) {$color = "#F6BE00";} // poor

```

```

if ($izi_st[$i] < 0.141) {$color = "FF0000";} // bad

if ($zone_type == 2) { echo "<font color ='#008000'>GES</font>, <font color ='#FF0000'>Not  
GES</font>";}
else {echo "<font color ='#008000'>EEL range is High</font>,  
<font color ='#54C571'>EEL range is Good</font>,  
<font color ='#F5E216'>EEL range is Moderate</font>,  
<font color ='#F6BE00'>EEL range is Poor</font>,  
<font color ='#FF0000'>EEL range is Bad</font> <br></br>"  
;}
}

$sql_meta = "SELECT [IDMesozoopMeta], Stations.idst, Stations.Name, UpperRange, LowerRange  

    FROM [emblas].[dbo].[MesozoopMeta]  

    inner join Stations on [MesozoopMeta].IDStationName = Stations.IDSt  

    where  

    IDStationName in (select idst from stations where idZone in ($id_zone))  

    $sql_date  

    order by IDMesozoopMeta";  

//echo $sql_meta; die;

$id_meta = sqlsrv_query($conn,$sql_meta,$params,$options);
    while($row = sqlsrv_fetch_array($id_meta,
SQLSRV_FETCH_ASSOC)){
        array_push($id_meta_arr, $row['IDMesozoopMeta']);
        array_push($st_name_arr, $row['Name']);
        array_push($horizons_arr, $row['UpperRange'] . "-" .  

$row['LowerRange']);
        //array_push($id_stations, $row['idst']);
        //echo $row['UpperRange'] . "-" . $row['LowerRange'];
    }

//print_r ($id_stations); echo "<br>";
//print_r ($st_name_arr); echo "<br>";

$row=null;
//print_r ($id_meta_arr);
//echo count($id_meta_arr);
$Total_bio = 0; $all_calc_bio = array();
for ($i = 0; $i < count($id_meta_arr); $i++ )
{
    //sql_station_id = "select "
}

$sql_total_bio = "Select IdMeta, SUM(StBiomass) as StBioAll, SUM(CalcBiomass) as CalcBioAll  

    from MesozoopSamples  

    where IDMeta in ($id_meta_arr[$i])  

    group by IDMeta  

    order by IDMeta";  

//echo ($i. " | ". $sql_total_bio . "<br>");

$total_bio = sqlsrv_query($conn,$sql_total_bio,$params,$options);

```

```

while($row = sqlsrv_fetch_array($total_bio, SQLSRV_FETCH_ASSOC))
{
    if ($row['CalcBioAll'] != null)
    {
        $bio_total = $row['CalcBioAll'];
    }
    else
    {
        $bio_total = $row['StBioAll'];
    }
}

$sql_all_bio_sample = "Select idMeta, StBiomass, CalcBiomass
from MesozoopSamples
where idmeta in ($id_meta_arr[$i])
order by IDMeta";

//echo $sql_all_bio_sample; echo "<br>";

$all_bio_sample = sqlsrv_query($conn,$sql_all_bio_sample,$params,$options);
while($row = sqlsrv_fetch_array($all_bio_sample, SQLSRV_FETCH_ASSOC))
{
    if ($row['CalcBiomass'] == null && $row['StBiomass'] == null) {array_push($all_calc_bio,
0);}

    if ($row['CalcBiomass'] != null)
    {
        $bio = log(($row['CalcBiomass'])/$bio_total),2)*($row['CalcBiomass'])/$bio_total);
        // $bio = (log($row['CalcBiomass']),
        2)/$total_bio)*($row['CalcBiomass'])/$total_bio); //echo $bio; echo "<br>";
        array_push($all_calc_bio, $bio);
    }

    if ($row['StBiomass'] != null && $row['CalcBiomass'] == null)
    {
        $bio = log(($row['StBiomass'])/$bio_total),2)*($row['StBiomass'])/$bio_total);
        // $bio = (log($row['StBiomass']), 2)/$total_bio)*($row['StBiomass'])/$total_bio);
//echo $bio; echo "<br>";
        array_push($all_calc_bio, $bio);
    }
}

$sql_copepoda_total = "SELECT [IDMeta]
,SUM([StBiomass]) as [StBiomass]
,SUM([CalcBiomass]) as [CalcBiomass]
FROM [emblas].[dbo].[MesozoopSamples]"

```

```

inner join MesozoopMeta on
[MesozoopSamples].IDMeta = MesozoopMeta.IDMesozoopMeta
inner join MesozoopSubName on
[MesozoopSamples].IDMesozoopSubName = MesozoopSubName.MesozoopSubNameID
inner join MesozoopName on
MesozoopSubName.MesozoopNameID = MesozoopName.MesozoopNameID
where
MesozoopSamples.IdMeta in ($id_meta_arr[$i])
and ([MesozoopName].MesozoopNameID in
(SELECT [MesozoopNameID] FROM [emblas].[dbo].[MesozoopName] where [MesozoopNameID] in (6,
7, 9)))
group by [IDMeta]";

$all_copepoda_total = sqlsrv_query($conn,$sql_copepoda_total,$params,$options);
while($row = sqlsrv_fetch_array($all_copepoda_total, SQLSRV_FETCH_ASSOC))
{
    if ($row['CalcBiomass'] == null && $row['StBiomass'] == null)
{array_push($total_copepoda_ind, 0);}

    if ($row['CalcBiomass'] != null)
    {
        $copepoda_total = $row['CalcBiomass']/$bio_total;
        array_push($total_copepoda_ind, $copepoda_total);
    }

    if ($row['StBiomass'] != null && $row['CalcBiomass'] == null)
    {
        $copepoda_total = $row['StBiomass']/$bio_total;
        array_push($total_copepoda_ind, $copepoda_total);
    }
}

$sql_noctiluka_total = "SELECT
[IDMeta]
,SUM([StBiomass]) as [StBiomass]
,SUM([CalcBiomass]) as [CalcBiomass]
FROM [emblas].[dbo].[MesozoopSamples]
where IDMesozoopSubName = 378
and IDMeta in ($id_meta_arr[$i])
group by [IDMeta]";

$all_noctiluka_total = sqlsrv_query($conn,$sql_noctiluka_total,$params,$options);
if (sqlsrv_num_rows($all_noctiluka_total) == 0)
{
    $noctiluka_total = 0;
}
else
{
    while($row = sqlsrv_fetch_array($all_noctiluka_total, SQLSRV_FETCH_ASSOC))
    {
        if ($row['CalcBiomass'] == null && $row['StBiomass'] == null) {$noctiluka_total =
0;}
        if ($row['CalcBiomass'] != null)

```

```

        {
            $noctiluka_total = $row['CalcBiomass']/$bio_total;
        }
        if ($row['StBiomass'] != null && $row['CalcBiomass'] == null)
        {
            $noctiluka_total = $row['StBiomass']/$bio_total;
        }
    }

}

$sql_jelly_total = "SELECT
    [IDMeta]
    ,SUM([StBiomass]) as [StBiomass]
    ,SUM([CalcBiomass]) as [CalcBiomass]
    FROM [emblas].[dbo].[MesozoopSamples]
    inner join MesozoopSubName on
[MesozoopSamples].IDMesozoopSubName = MesozoopSubName.MesozoopSubNameID
    inner join MesozoopName on
MesozoopSubName.MesozoopNameID = MesozoopName.MesozoopNameID
    where MesozoopName.MesozoopNameID = 2
    and IDMeta in ($id_meta_arr[$i])
    group by [IDMeta]";

//echo $sql_jelly_total; die;

$all_jelly_total = sqlsrv_query($conn,$sql_jelly_total,$params,$options);
if (sqlsrv_num_rows($all_jelly_total) == 0)
{
    $jelly_total = 0;
}
else
{
    while($row = sqlsrv_fetch_array($all_jelly_total, SQLSRV_FETCH_ASSOC))
    {
        if ($row['CalcBiomass'] == null && $row['StBiomass'] == null) {$jelly_total = 0;}
        if ($row['CalcBiomass'] != null)
        {
            $jelly_total = $row['CalcBiomass']/$bio_total;
        }
        if ($row['StBiomass'] != null && $row['CalcBiomass'] == null)
        {
            $jelly_total = $row['StBiomass']/$bio_total;
        }
    }
}

$H = array_sum($all_calc_bio) * (-1);

if ($bio_total != "<b>NULL</b>")
{
    $total_bio_index = 0.894 * pow($bio_total,(-0.006));
}
```

```

        array_push($total_bio_ind, $total_bio_index);
    }
    else
    {
        $total_bio_index = "<b>NULL</b>";
        array_push($total_bio_ind, null);
    }

    if ($H != "<b>NULL</b>")
    {
        $H_index = $H / 3.5;
        array_push($total_shannon_ind, $H_index);
    }
    else
    {
        $H_index = "<b>NULL</b>";
        array_push($total_shannon_ind, null);
    }

    $noctiluka_total_index = 1 - $noctiluka_total;
    array_push($total_noctiluca_ind, $noctiluka_total_index);

    $jelly_total_index = 1 - $jelly_total;
    array_push($total_jelly_ind, $jelly_total_index);

// $noctiluka_total_index = 1 - $noctiluka_total;
// $jelly_total_index = 1 - $jelly_total;

array_push($izi_calc, $total_bio_ind[$i]);
array_push($izi_calc, $total_shannon_ind[$i]);
array_push($izi_calc, $total_copepoda_ind[$i]);
array_push($izi_calc, $total_noctiluca_ind[$i]);
array_push($izi_calc, $total_jelly_ind[$i]);

//print_r ($izi_calc); echo "<br>";
// $izi = min($izi_calc);

$index = array_search(min($izi_calc), $izi_calc); // echo $izi_calc[$index]; echo "<br>";

if ($index == 0){$izi = pow(pow($izi_calc[0],0.78),0.5) *
pow(pow($izi_calc[1],0.62)*pow($izi_calc[2],0.87)*pow($izi_calc[3],0.9)*pow($izi_calc[4],0.36),2.5);}
if ($index == 1){$izi = pow(pow($izi_calc[1],0.62),0.5) *
pow(pow($izi_calc[0],0.78)*pow($izi_calc[2],0.87)*pow($izi_calc[3],0.9)*pow($izi_calc[4],0.36),2.5);}
if ($index == 2){$izi = pow(pow($izi_calc[2],0.87),0.5) *
pow(pow($izi_calc[0],0.78)*pow($izi_calc[1],0.62)*pow($izi_calc[3],0.9)*pow($izi_calc[4],0.36),2.5);}
if ($index == 3){$izi = pow(pow($izi_calc[3],0.9),0.5) *
pow(pow($izi_calc[0],0.78)*pow($izi_calc[1],0.62)*pow($izi_calc[2],0.87)*pow($izi_calc[4],0.36),2.5);}
if ($index == 4){$izi = pow(pow($izi_calc[4],0.36),0.5) *
pow(pow($izi_calc[0],0.78)*pow($izi_calc[1],0.62)*pow($izi_calc[2],0.87)*pow($izi_calc[3],0.9),2.5);}

//echo min($izi_calc);

```

```

echo "<tr>
    <td>". $id_meta_arr[$i] . "</td>
    <td>" . $st_name_arr[$i]. "</td>
    <td>" . $horisons_arr[$i] . "</td>
    <td>" . $bio_total . "</td>
    <td>" . $H . "</td>
    <td>" . $scopepoda_total ."</td>
    <td>" . $noctiluka_total . "</td>
    <td>" . $jelly_total . "</td>
    <td>" . $total_bio_index . "</td>
    <td>" . $H_index . "</td>
    <td>" . $scopepoda_total . "</td>
    <td>" . $noctiluka_total_index . "</td>
    <td>" . $jelly_total_index . "</td>
    ";
}

//array_push($izi_stations, $id_stations[$i] => $izi);
array_push($izi_stations, $izi);

if ($zone_type == 2)
{
    if ($izi > 0.543) { $color = "#008000";}
    if ($izi < 0.542) { $color = "#FF0000";}
}
else
{
    if ($izi > 0.844) {$color = "008000";} // hight
    if ($izi < 0.844 && $izi > 0.543) {$color = "#54C571";} // good
    if ($izi < 0.542 && $izi > 0.264) {$color = "#F5E216";} // moderate
    if ($izi < 0.263 && $izi > 0.141) {$color = "#F6BE00";} // poor
    if ($izi < 0.141) {$color = "FF0000";} // bad
}

//$izi_stations[] = [$id_stations[$i] => $izi];

echo "<td bgcolor = '$color' width = '100'>" . $izi. "</td>";

//<td>" . $izi. "</td>
echo "</tr>";

$all_calc_bio = array(); $H = null; $bio = 0;$izi_calc = array(); $izi = null;
}
echo "</table></center>";

$izi_st = array();

$key = $st_name_arr[0]; $value = $izi_stations[0]; $count = 1;

for ($i = 1; $i <= (count($izi_stations)+1); $i++)

```

```

{
    if ($st_name_arr[$i] == $key)
    {
        $count++; //= $count + 1;
        $value += $izi_stations[$i];
        $key = $st_name_arr[$i];
        //echo $key; echo "<br>";
    }

    if ($st_name_arr[$i] != $key)
    {

        //echo "Count $count Value ".round($value, 4) ." key $key <br>";
        //echo $value; echo " | "; echo round(($value/$count),4);die;
        array_push ($izi_st, ($value/$count));
        array_push ($id_station_names, $key);
        $value = $izi_stations[$i]; $count = 1; $key = $st_name_arr[$i];
    }
    //array_push ($izi_st, round(($value/$count),4));
    //array_push ($id_station_names, $key);
}
//echo "Count $count"; echo "<br>";
//echo "Value $value"; echo "<br>"; //echo "test";

echo "<br><br><br><br><br><center>";
echo "<table border = '1'>
<tr>
    <th>Station Name</th>
    <th>Integral Zooplankton Index</th>
</tr>
";
for ($i = 0; $i < count($izi_st); $i++)
{
    if ($zone_type == 2)
    {
        if ($izi_st[$i] > 0.543) { $color = "#008000";}
        if ($izi_st[$i] < 0.542) { $color = "#FF0000";}
    }
    else
    {
        if ($izi_st[$i] > 0.844) {$color = "008000"} // hight
        if ($izi_st[$i] < 0.844 && $izi_st[$i] > 0.543) {$color = "#54C571"} // good
        if ($izi_st[$i] < 0.542 && $izi_st[$i] > 0.264) {$color = "#F5E216"} // moderate
        if ($izi_st[$i] < 0.263 && $izi_st[$i] > 0.141) {$color = "#F6BE00"} // poor
        if ($izi_st[$i] < 0.141) {$color = "FF0000"} // bad
    }
    echo "<tr><td>" . $id_station_names[$i]. "</td>";
    echo "<td bgcolor = '$color'>" . $izi_st[$i]. "</td>";

    echo "</tr>";
}

```

```

echo "</table> <br><br><br><br>";

$total_izi_polygon = array_sum($izi_st)/count($izi_st);

if ($zone_type == 2)
{
    if ($total_izi_polygon > 0.543) { $color = "#008000"; echo "<H1><p style='color: $color'>$zone_name (Open Water) GES " . $total_izi_polygon . "</p>";}
        if ($total_izi_polygon < 0.542) { $color = "#FF0000"; echo "<H1><p style='color: $color'>$zone_name (Open Water) Not GES " . $total_izi_polygon . "</p>);}

    }
else
{
    if ($total_izi_polygon > 0.844)
    {
        $color = "008000";echo "<H1><p style='color: $color'> ELL range for
$zone_name (Marine coastal waters) is High " . $total_izi_polygon . "</p>";
    } // hight
        if ($total_izi_polygon < 0.844 && $total_izi_polygon > 0.543)
            {$color = "#54C571";echo "<H1><p style='color: $color'> ELL range for
$zone_name (Marine coastal waters) is Good " . $total_izi_polygon . "</p>";} // good
            if ($total_izi_polygon < 0.542 && $total_izi_polygon > 0.264) {$color =
"#F5E216";echo "<H1><p style='color: $color'> ELL range for $zone_name (Marine coastal waters) is
Moderate " . $total_izi_polygon . "</p>";} // moderate
                if ($total_izi_polygon < 0.263 && $total_izi_polygon > 0.141) {$color =
"#F6BE00";echo "<H1><p style='color: $color'> ELL range for $zone_name (Marine coastal waters) is
Poor " . $total_izi_polygon . "</p>";} // poor
                    if ($total_izi_polygon < 0.141) {$color = "FF0000";echo "<H1><p style='color: $color'> ELL range for $zone_name (Marine coastal waters) is Bad " . $total_izi_polygon . "</p>";} // bad
    }
}

?>
<?php

$par_gr = $_POST['par_gr'];
$id_par = $_POST['id_par'];
$id_loc = $_POST['id_loc'];
$loc = $_POST['loc'];
$date_period = $_POST['date_period'];
$date_begin = $_POST['date_begin'];
$date_end = $_POST['date_end'];
$month = $_POST['month'];
$year = $_POST['year'];

session_start();
require_once ($_SERVER['DOCUMENT_ROOT']."/connectionEmblas.php");
$params = array();
$options = array( "Scrollable" => SQLSRV_CURSOR_KEYSET );

```

```

if ($date_period == "date")
{
    $begin = explode(".", $date_begin);
    $begin_for_sql = $begin[2] . "-" . $begin[1] . "-" . $begin[0]; //echo $begin_for_sql;echo "</br>";
    $end = explode(".", $date_end);
    $end_for_sql = $end[2] . "-" . $end[1] . "-" . $end[0];
    $sql_date = " and (CONVERT (date,(Convert (varchar,Stations.DayStart) + .' + Convert
(varchar,Stations.MonthStart) + .' + Convert (varchar,Stations.YearStart)), 104) between '$begin_for_sql'
and '$end_for_sql')"; //echo($sql_date);
}
if ($date_period == "period")
{
    $m = implode("", $month); //echo "period";
    $y = implode("", $year);
    $sql_date = " and (Stations.MonthStart in ($m) and Stations.YearStart in ($y))";
}
//echo $sql_date; //die; //echo "</br>";

if ($loc == "station")
{
    $sql_part = "Select Stations.IDst as St, Stations.Name as Name, LonDecimalStart as Longitude,
LatDecimalStart as Latitude,
('<b>Cruise:</b> ' + DataSource.Name + ', <b>Station:</b> ' + Stations.Name) as Description
,CONVERT (date,(Convert (varchar,Stations.DayStart) + .' + Convert
(varchar,Stations.MonthStart) + .' + Convert (varchar,Stations.YearStart)), 104) as 'DateStart'
from Stations
inner join DataSource on Stations.IDdataSource = DataSource.IDdataSource
where Stations.IDdataSource in ($id_loc) $sql_date";
}

//echo $sql_part; die;

if ($loc == "zone")
{
    $sql_part = "select Stations.IDst as St, Stations.Name as [Name], LonDecimalStart as Longitude,
LatDecimalStart as Latitude,'<b>Cruise:</b> ' + DataSource.Name + ', <b>Station:</b> ' +
Stations.Name) as Description
,CONVERT (date,(Convert (varchar,Stations.DayStart) + .' + Convert
(varchar,Stations.MonthStart) + .' + Convert (varchar,Stations.YearStart)), 104) as 'DateStart'
from Stations
inner join MetaHydro on Stations.IDst = MetaHydro.IDStationName
inner join SamplesHydro on MetaHydro.IDMetaHydro = SamplesHydro.IDMetaHydro
inner join Parameter on SamplesHydro.IDpar = Parameter.IDpar
inner join ParameterName on Parameter.IDparName = ParameterName.IDparName
inner join DataSource on Stations.IDdataSource = DataSource.IDdataSource
where
Stations.idZone in ($id_loc)
and ParameterName.IDparGroup in ($par_gr)
and Parameter.IDpar in ($id_par)
$sql_date
group by Stations.idSt, Stations.Name, LonDecimalStart, LatDecimalStart, ('<b>Cruise:</b> ' +
DataSource.Name + ', <b>Station:</b> ' + Stations.Name)

```

```

,CONVERT (date,(Convert (varchar,Stations.DayStart)+'.'+Convert
(varchar,Stations.MonthStart)+'.'+Convert (varchar,Stations.YearStart)), 104)
order by Stations.idSt";

}

//echo $sql_part; die;

$stations = sqlsrv_query($conn,$sql_part,$params,$options);

$data = "";

//echo $staions;
//echo $sql_part; die;

while($row = sqlsrv_fetch_array($stations,
SQLSRV_FETCH_ASSOC))
{
    $par_name = "";
    $hor = "/// //echo "Station ".$row['St'] . " | <br>";
    $sql_hor = "SELECT
[MetaHydro].[IDStationName],[MetaHydro].[SamplingDepth],ParameterName.Name
FROM
[emblas].[dbo].[MetaHydro]
inner join SamplesHydro on
MetaHydro.IDMetaHydro = SamplesHydro.IDMetaHydro
inner join Parameter on
SamplesHydro.IDpar = Parameter.IDpar
inner join ParameterName on
Parameter.IDparName = ParameterName.IDparName
where
SamplesHydro.IDpar in ($id_par)
and MetaHydro.IDStationName
in (". $row['St'] .")
group by
[MetaHydro].[IDStationName], [MetaHydro].[SamplingDepth], ParameterName.Name
order by
[MetaHydro].[IDStationName]";

    //echo $sql_hor; die;

    $horizons =
sqlsrv_query($conn,$sql_hor,$params,$options); //echo $horizons;
    while($row1 = sqlsrv_fetch_array($horizons,
SQLSRV_FETCH_ASSOC))
    {
        $hor .= $row1['SamplingDepth'] . ", ";
        $par_name = $row1['Name'];
    }

    $hor_2 = substr_replace($hor, "", -2);
    $date = $row['DateStart']->format('d.m.Y');
}

```

```

        $data .= "{\"title\": \" \"$row['Name'] .\" ,\"lat\": \""
        ".$row['Latitude'] .\",\"lng\": \" \"$row['Longitude'] .\" ,\"description\": \""
                    .$row['Description'] .", <b>Horizons for $par_name:
</b> \"$hor_2 .\", <b>Date: </b> . $date . \" ,\"color\": \"red-dot\"};";
    }

    $data_2 = substr_replace($data, "", -1);
    echo "[" . $data_2 ."]";

?>

<?php
$id = $_POST['id'];

    session_start();
    require_once ($_SERVER['DOCUMENT_ROOT']."/connectionEmblas.php");
    $params = array();
    $options = array( "Scrollable" => SQLSRV_CURSOR_KEYSET );

if ($id == 1) {//General physico-chemical parameters & Nutrients
    $sql_par = "select ParameterGroup.IDparGroup, ParameterGroup.Name
from SamplesHydro
inner join Parameter on SamplesHydro.IDpar = Parameter.IDpar
inner join ParameterName on Parameter.IDparName = ParameterName.IDparName
inner join ParameterGroup on ParameterName.IDparGroup = ParameterGroup.IDparGroup
group by ParameterGroup.IDparGroup, ParameterGroup.Name";
}
if ($id == 2) {//Pollution in water
    $sql_par = "select ParameterGroup.IDparGroup, ParameterGroup.Name
from [SamplesWater]
inner join Parameter on [SamplesWater].IDpar = Parameter.IDpar
inner join ParameterName on Parameter.IDparName = ParameterName.IDparName
inner join ParameterGroup on ParameterName.IDparGroup = ParameterGroup.IDparGroup
group by ParameterGroup.IDparGroup, ParameterGroup.Name";
}
if ($id == 3) {//Pollution in bottom
$sql_par = "select ParameterGroup.IDparGroup, ParameterGroup.Name
from [SamplesSediment]
inner join Parameter on [SamplesSediment].IDpar = Parameter.IDpar
inner join ParameterName on Parameter.IDparName = ParameterName.IDparName
inner join ParameterGroup on ParameterName.IDparGroup = ParameterGroup.IDparGroup
group by ParameterGroup.IDparGroup, ParameterGroup.Name";
}
if ($id == 4) {//Pollution in biota
$sql_par = "select ParameterGroup.IDparGroup, ParameterGroup.Name
from [SamplesBiota]
inner join Parameter on [SamplesBiota].IDpar = Parameter.IDpar
inner join ParameterName on Parameter.IDparName = ParameterName.IDparName
inner join ParameterGroup on ParameterName.IDparGroup = ParameterGroup.IDparGroup
group by ParameterGroup.IDparGroup, ParameterGroup.Name";
}

//echo $sql_par;

```

```

$parametrs = sqlsrv_query($conn,$sql_par,$params,$options);

$data = "";

while($row = sqlsrv_fetch_array($parametrs,
SQLSRV_FETCH_ASSOC))
{
    //echo $row['SamplingDepth'];
    $data .= "{$id_par}: \"$row['IDparGroup']\" ,\"par_name\": \"$row['Name']"
."\"};";
}

$data_2 = substr_replace($data, "", -1);
echo $data_2;

?>

<?php

$par_gr = $_POST['par_gr'];
$matrix = $_POST['matrix'];
$datasource_id = $_POST['datasource'];
$loc = $_POST['loc'];

session_start();
require_once ($_SERVER['DOCUMENT_ROOT']."/connectionEmblas.php");
$params = array();
$options = array( "Scrollable" => SQLSRV_CURSOR_KEYSET );

if ($loc == "station")
{
    if ($matrix == 1)
    {
        $sql_par = "SELECT
Parameter.[IDpar]
,ParameterName.Name

FROM [emblas].[dbo].[SamplesHydro]
inner join MetaHydro on SamplesHydro.IDMetaHydro = MetaHydro.IDMetaHydro
inner join Stations on MetaHydro.IDStationName = Stations.IDst
inner join DataSource on Stations.IDdataSource = DataSource.IDdataSource
inner join Parameter on [SamplesHydro].IDpar = Parameter.IDpar
inner join ParameterName on Parameter.IDparName = ParameterName.IDparName
inner join ParameterGroup on ParameterName.IDparGroup = ParameterGroup.IDparGroup
where DataSource.IDdataSource in ($datasource_id)
and ParameterGroup.IDparGroup in ($par_gr)
group by Parameter.[IDpar],ParameterName.Name
order by ParameterName.Name";
    }
}

if ($matrix == 2)
{

```

```

$sql_par = "SELECT
Parameter.[IDpar]
,ParameterName.Name

FROM [emblas].[dbo].SamplesWater
inner join MetaPolWater on SamplesWater.IDMetaPolWater = MetaPolWater.IDMetaPolWater
inner join Stations on MetaPolWater.IDStationName = Stations.IDst
inner join DataSource on Stations.IDdataSource = DataSource.IDdataSource
inner join Parameter on SamplesWater.IDpar = Parameter.IDpar
inner join ParameterName on Parameter.IDparName = ParameterName.IDparName
inner join ParameterGroup on ParameterName.IDparGroup = ParameterGroup.IDparGroup
where DataSource.IDdataSource in ($datasource_id)
and ParameterGroup.IDparGroup in ($par_gr)
group by Parameter.[IDpar],ParameterName.Name
order by ParameterName.Name";
}

if ($matrix == 3)
{
    $sql_par = "SELECT
Parameter.[IDpar]
,ParameterName.Name
        FROM [emblas].[dbo].[SamplesSediment]
inner join MetaPolSed on SamplesSediment.IDMetaPolSed = MetaPolSed.IDMetaPolSed
inner join Stations on MetaPolSed.IDStationName = Stations.IDst
inner join DataSource on Stations.IDdataSource = DataSource.IDdataSource
inner join Parameter on SamplesSediment.IDpar = Parameter.IDpar
inner join ParameterName on Parameter.IDparName = ParameterName.IDparName
inner join ParameterGroup on ParameterName.IDparGroup = ParameterGroup.IDparGroup
where DataSource.IDdataSource in ($datasource_id)
and ParameterGroup.IDparGroup in ($par_gr)
group by Parameter.[IDpar],ParameterName.Name
order by ParameterName.Name";
}

if ($matrix == 4)
{
    $sql_par = "SELECT
Parameter.[IDpar]
,ParameterName.Name

FROM [emblas].[dbo].SamplesBiota
inner join MetaPolBiota on SamplesBiota.IDMetaPolBiota = MetaPolBiota.IDMetaPolBiota
inner join Stations on MetaPolBiota.IDStationName = Stations.IDst
inner join DataSource on Stations.IDdataSource = DataSource.IDdataSource
inner join Parameter on SamplesBiota.IDpar = Parameter.IDpar
inner join ParameterName on Parameter.IDparName = ParameterName.IDparName
inner join ParameterGroup on ParameterName.IDparGroup = ParameterGroup.IDparGroup
where DataSource.IDdataSource in ($datasource_id)
and ParameterGroup.IDparGroup in ($par_gr)
group by Parameter.[IDpar],ParameterName.Name
order by ParameterName.Name";
}
}

```

```

if ($loc == "zone")
{
    if ($matrix == 1)
    {
        $sql_par = "SELECT
Parameter.[IDpar]
,ParameterName.Name

FROM [emblas].[dbo].[SamplesHydro]
inner join MetaHydro on SamplesHydro.IDMetaHydro = MetaHydro.IDMetaHydro
inner join Stations on MetaHydro.IDStationName = Stations.IDst
inner join Parameter on [SamplesHydro].IDpar = Parameter.IDpar
inner join ParameterName on Parameter.IDparName = ParameterName.IDparName
inner join ParameterGroup on ParameterName.IDparGroup = ParameterGroup.IDparGroup
where Stations.idZone in ($datasource_id)
and ParameterGroup.IDparGroup in ($par_gr)
group by Parameter.[IDpar],ParameterName.Name
order by ParameterName.Name";
    }

if ($matrix == 2)
{
    $sql_par = "SELECT
Parameter.[IDpar]
,ParameterName.Name

FROM [emblas].[dbo].SamplesWater
inner join MetaPolWater on SamplesWater.IDMetaPolWater = MetaPolWater.IDMetaPolWater
inner join Stations on MetaPolWater.IDStationName = Stations.IDst
inner join Parameter on SamplesWater.IDpar = Parameter.IDpar
inner join ParameterName on Parameter.IDparName = ParameterName.IDparName
inner join ParameterGroup on ParameterName.IDparGroup = ParameterGroup.IDparGroup
where Stations.idZone in ($datasource_id)
and ParameterGroup.IDparGroup in ($par_gr)
group by Parameter.[IDpar],ParameterName.Name
order by ParameterName.Name";
}

if ($matrix == 3)
{
    $sql_par = "SELECT
Parameter.[IDpar]
,ParameterName.Name

        FROM [emblas].[dbo].[SamplesSediment]
inner join MetaPolSed on SamplesSediment.IDMetaPolSed = MetaPolSed.IDMetaPolSed
inner join Stations on MetaPolSed.IDStationName = Stations.IDst
inner join Parameter on SamplesSediment.IDpar = Parameter.IDpar
inner join ParameterName on Parameter.IDparName = ParameterName.IDparName
inner join ParameterGroup on ParameterName.IDparGroup = ParameterGroup.IDparGroup
where Stations.idZone in ($datasource_id)
and ParameterGroup.IDparGroup in ($par_gr)
group by Parameter.[IDpar],ParameterName.Name
order by ParameterName.Name";
}

```

```

}

if ($matrix == 4)
{
    $sql_par = "SELECT
Parameter.[IDpar]
,ParameterName.Name

FROM [emblas].[dbo].SamplesBiota
inner join MetaPolBiota on SamplesBiota.IDMetaPolBiota = MetaPolBiota.IDMetaPolBiota
inner join Stations on MetaPolBiota.IDStationName = Stations.IDSt
inner join Parameter on SamplesBiota.IDpar = Parameter.IDpar
inner join ParameterName on Parameter.IDparName = ParameterName.IDparName
inner join ParameterGroup on ParameterName.IDparGroup = ParameterGroup.IDparGroup
where Stations.idZone in ($datasource_id)
and ParameterGroup.IDparGroup in ($par_gr)
group by Parameter.[IDpar],ParameterName.Name
order by ParameterName.Name";
}

}

$parametrs = sqlsrv_query($conn,$sql_par,$params,$options);

$data = "";

while($row = sqlsrv_fetch_array($parametrs,
SQLSRV_FETCH_ASSOC))
{
    //echo $row['SamplingDepth'];
    $data .= "{\"id_par\": \"$row['IDpar']\", \"par_name\": \"$row['Name']\"},";
}

$data_2 = substr_replace($data, "", -1);
echo $data_2;

?>

window.onload = function () {
    var mapOptions = {
        center: new google.maps.LatLng(44, 34),
        zoom: 5,
        mapTypeId: google.maps.MapTypeId.ROADMAP
    };
    map = new google.maps.Map(document.getElementById("dvMap"), mapOptions);
}

$(document).ready(function () {

    var today = new Date();
    var curYear = today.getFullYear();
    $('#datebegin1').datepicker({
        // defaultDate: "+1w",

```

```

changeMonth: true,
changeYear: true,
yearRange: "1999:"+curYear,
numberOfMonths: 1,
// showButtonPanel:
dateFormat:
altField:
altFormat: "DD, d MM,
showOn: "button",
buttonImage:
buttonImageOnly: true,
buttonText: "Select
onSelect: function(
$( "#dateend"
).datepicker( "option", "minDate", selectedDate );
//$('#alternateDP_fromHide').val(selectedDate);
}
});
$( "#dateend1" ).datepicker({
// defaultDate: "+1w",
changeMonth: true,
changeYear: true,
yearRange: "1999:"+curYear,
numberOfMonths: 1,
showOn: "button",
buttonImage:
buttonImageOnly: true,
buttonText: "Select
// showButtonPanel:
dateFormat:
altField:
altFormat: "DD, d MM,
onSelect: function(
$( "#datebegin"
).datepicker( "option", "maxDate", selectedDate );
//$('#alternateDP_toHide').val(selectedDate);
}
});

```

```

        $(this).change());
    });

    $("#dateend1").change( function() {
        var end = $("#dateend1").val();
        var begin = $("#datebegin1").val();
        var id_zone = $("#zone").val();
        //alert('Par group | ' + id_par_gr);
        //var matrix_id = $("#matrix").val(); //alert ('Matrix | ' + matrix_id);
        //alert(date_period);

        $.ajax({
            url: "map1.php",
            type: "post",
            data:{id: id_zone, loc: id_zone, date_begin: begin, date_end: end},
            success: function(result){
                show_stations(result);
            }
        });
    });

    var starYear = 1999; var i = 1;
    do {

        $("#year").append('<option value="' + starYear +
        '">' + starYear + '</option>');
        i++;
        starYear++; //alert (starYear);
    } while (starYear <= curYear)

    $("#datebegin").datepicker({
        // defaultDate: "+1w",
        changeMonth: true,
        changeYear: true,
        yearRange:
        "1999:" + curYear,
        numberOfMonths: 1,
        // showButtonPanel:
        true,
        dateFormat:
        "dd.mm.yy",
        altField:
        "#alternateDP_from",
        altFormat: "DD, d MM,
        yy",
        showOn: "button",
        buttonImage:
        "image/calendar.gif",
        buttonImageOnly: true,
    });
}

```

```

buttonText: "Select
date",
selectedDate ) {
    //$('#alternateDP_fromHide').val(selectedDate);
}
});
$( "#dateend" ).datepicker({
// defaultDate: "+1w",
changeMonth: true,
changeYear: true,
yearRange:
"1999:"+curYear,
"image/calendar.gif",
date",
true,
"dd.mm.yy",
"#alternateDP_to",
yy",
selectedDate ) {
.datepicker( "option", "maxDate", selectedDate );
//$('#alternateDP_toHide').val(selectedDate);
}
});

$( "#dateend").change( function() {
var end = $("#dateend").val();
var begin = $("#datebegin").val();
var id_par_gr = $("#par_group").val(); //alert('Par group | ' + id_par_gr);
//var matrix_id = $("#matrix").val(); //alert ('Matrix | ' + matrix_id);
var id_par = $("#par").val(); //alert('Parameter | ' + id_par);
var id_loc = $("#location").val(); //alert ('Cruise | ' + id_loc);
var location =
document.querySelector('input[name="zone_stations"]:checked').value; //alert (location);
}
);

```

```

        var date_period =
document.querySelector('input[name="date_period"]:checked').value;
//alert(date_period);

$.ajax({
    url: "map.php",
    type: "post",
    data:{par_gr: id_par_gr, id_par: id_par, id_loc: id_loc, loc: location,
date_period: date_period, date_begin: begin, date_end: end},
    success: function(result){
        show_stations(result);
    }
});

});

$("#year").change( function() {

    var y = $("#year").val();
    var m = $("#month").val();
    //alert(y);
    //alert(m);
    var id_par_gr = $("#par_group").val(); //alert('Par group | ' + id_par_gr);
    //var matrix_id = $("#matrix").val(); //alert ('Matrix | ' + matrix_id);
    var id_par = $("#par").val();
    var id_loc = $("#location").val(); //alert ('Cruise | ' + datasource_id);
    var location =
document.querySelector('input[name="zone_stations"]:checked').value; //alert (location);
    var date_period =
document.querySelector('input[name="date_period"]:checked').value;
//alert(date_period);

    $.ajax({
        url: "map.php",
        type: "post",
        data:{par_gr: id_par_gr, id_par: id_par, id_loc: id_loc, loc: location,
date_period: date_period, month: m, year: y},
        success: function(result){
            show_stations(result);
        }
    });

});

$("#par").change(function(){
    var pr = $("#par").val();
    //alert (pr);
});

$('input[type=radio][name="zone_stations"]').on('change', function() {

```

```

//alert(this.value);

    if (this.value == "station") {
        $("#location").show();
        $("#label_for_location").show();
        $("#layer_type").hide();

        $.ajax({
            url:"location.php", //the page containing php script
            type: "post", //request type,
            data: {id: 1},
            success:function(result){
                $("#location").empty();$("#location").append('<option value="none" selected disabled hidden>Select a Cruise</option>');
                var stations = JSON.parse('[' + result + ']');
                for (i = 0; i < stations.length; i++) {
                    var data = stations[i];
                    $("#label_for_location").append('Select Cruise, please');
                    $("#location").append('<option value="' + data.id_datasource +
                    '>' + data.ds_name + '</option>');
                };
            }
        });

    }

    if (this.value == "zone") {
        $("#location").show();
        $("#label_for_location").show();
        $("#layer_type").hide();

        $.ajax({
            url:"location.php", //the page containing php script
            type: "post", //request type,
            data: {id: 2},
            success:function(result){
                $("#location").empty();$("#location").append('<option value="none" selected disabled hidden>Select a Zone</option>');
                var stations = JSON.parse('[' + result + ']');
                for (i = 0; i < stations.length; i++) {
                    var data = stations[i];
                    $("#location").append('<option value="' + data.id_zone + '">' +
                    data.zone + '</option>');
                };
            }
        });

    }
});
```

```

        $("#" + matrix).change(function () {
            loc = document.querySelector('input[name="zone_stations"]:checked').value;
            //alert (loc);

            $("#dep").show();
            var matrix_id = $("#" + matrix).val(); //alert(matrix_id);

            if (matrix_id == 1 && loc == "station") {
                $("#layer_type").show();

                var id_cruise = $("#location").val();

                $.ajax({
                    url:"depth.php", //the page containing php script
                    type: "post", //request type,
                    data: {id: id_cruise},
                    success:function(result){
                        $("#layer_type").empty();$("#layer_type").append('<option value="none" selected disabled hidden>Select a Layer Type</option>');
                        var depths = JSON.parse('[' + result + ']');
                        //alert ( id_cruise + ' | ' + result);
                        for (i = 0; i < depths.length; i++) {
                            var data = depths[i];
                            $("#layer_type").append('<option value="' + data.id_layer + '">' + data.layer_name + '</option>');
                        };
                    }
                });
            }

            if (matrix_id == 2){
                $("#layer_type").hide();
                $("#layer_type").empty();
            }

            if (matrix_id > 2) {
                $("#dep").hide();
            }

            $.ajax({
                url:"matrix.php", //the page containing php script
                type: "post", //request type,
                data: {id: matrix_id},
                success:function(result){
                    $("#par_group").empty();
                    $("#par").empty(); $("#par").append('<option value="none" selected disabled hidden>Select a Parametr group first</option>');
                    var parametrs = JSON.parse('[' + result + ']');

```

```

        $($("#par_group").empty());$("#par_group").append('<option value="none" selected disabled hidden>Select a Matrix first</option>');
        for (i = 0; i < parametrs.length; i++) {
            var data = parametrs[i]; //alert (data.id_par + "" +
data.par_name)
            $("#par_group").append('<option value="' + data.id_par + '">' +
data.par_name + '</option>');
        };
    });
};

$("#par_group").change(function ()
{
    var id_par_gr = $("#par_group").val(); //alert('Par group | ' + id_par_gr);
    var matrix_id = $("#matrix").val(); //alert ('Matrix | ' + matrix_id);
    var datasource_id = $("#location").val(); //alert ('Cruise | ' + datasource_id);
    var location = document.querySelector('input[name="zone_stations"]:checked').value;
//alert (location);

    $.ajax({
        url:"parametr.php", //the page containing php script
        type: "post", //request type,
        data: {par_gr: id_par_gr, matrix: matrix_id, datasource: datasource_id, loc: location},
        success:function(result){

            $("#par").empty();
            var parametrs = JSON.parse('[ ' + result + ']');
            //alert (parametrs.length);
            $("#par").append('<option value="none" selected disabled hidden>No parameters
found</option>');
            if (parametrs.length > 0)
            {
                $("#par").empty();
                for (i = 0; i < parametrs.length; i++)
                {
                    var data = parametrs[i];
                    $("#par").append('<option value="' + data.id_par + '">' + data.par_name
+ '</option>');
                }
            };
        }
    });
});

$("#location").change(function ()
{
    var id_cruise = $("#location").val();

    $.ajax({
        url:"depth.php", //the page containing php script
        type: "post", //request type,

```

```
data: {id: id_cruise},
success:function(result){
    $("#layer_type").empty();$("#layer_type").append('<option
value="none" selected disabled hidden>Select a Layer Type</option>');
    var depths = JSON.parse(['' + result + ''']);
    //alert ( id_cruise + ' | ' + result);
    for (i = 0; i < depths.length; i++) {
        var data = depths[i];
        $("#layer_type").append('<option value="' + data.id_layer + '">' +
+ data.layer_name + '</option>');
    }
});
```

Додаток Б

«Код інтерактивної системи аналізу»

```
<!DOCTYPE
HTML>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0,
maximum-scale=1.0, user-scalable=no" />
<meta name="format-detection" content="telephone=no">
<meta http-equiv="X-UA-Compatible" content="IE=EDGE" />
<title>ArcGIS Web Application</title>
<link rel="shortcut icon" href="images/shortcut.ico">
<!--[if IE 8]>
<link rel="stylesheet" type="text/css" href="jimu.js/css/jimu-ie.css" />
<![endif]-->
<style type="text/css">
* {
    box-sizing: border-box;
}
body,
html {
    width: 100%;
    height: 100%;
    margin: 0;
    padding: 0;
    overflow: hidden;
}
#main-loading {
    width: 100%;
    height: 100%;
    text-align: center;
    overflow: hidden;
}
#main-page {
    display: none;
    width: 100%;
    height: 100%;
    position: relative;
}
```

```
#jimu-layout-manager {  
    width: 100%;  
    height: 100%;  
    position: absolute;  
}  
  
#unsupported-browser-message{  
    width: 100%;  
    height: 100%;  
    position: absolute;  
    left: 0;  
    top: 0;  
    background: rgba(0,0,0,0.6);  
    display: none;  
    align-items: center;  
    justify-content: center;  
    z-index: 1000;  
}  
  
.browser-message {  
    width: 700px;  
    max-width: 100%;  
    max-height: 100%;  
    overflow-y: auto;  
    background: #fff;  
    color: #333;  
    text-align: center;  
    padding: 1rem;  
}  
  
.browser-button-container {  
    display: flex;  
    justify-content: flex-end;  
}  
  
.browser-button-container button{  
    cursor: pointer;  
    border: none;  
    background: transparent;  
}  
  
.browser-button-container img{  
    width: 20px;  
    height: 20px;  
}  
  
.browser-message-title {  
    color: #002625;  
    font-weight: 300;
```

```
margin: 1.5rem auto;
font-size: 2.5rem;
line-height: 2rem;
}

.browser-message-content {
color: #3d6665;
font-weight: 400;
font-size: 1.1rem;
line-height: 2rem;
}

#browser-logo-list {
display: flex;
align-items: center;
margin: 2rem 1rem 0 1rem;
padding: 0;
}

.browser-logo {
display: inline-block;
margin: 0 auto;
}

.browser-logo img {
width: 3rem;
height: auto;
}

@media (min-width:700px) {
.browser-logo img {
width: 4rem;
}
}

</style>
<link rel="stylesheet" type="text/css" href="configs/loading/loading.css" />
<script>
*****
* This is the function you can modify to customize the loading page
* This function will be invoked when one resource is loaded.
*****/
// var progress;
// function loadingCallback(url, i, count) {
// var loading = document.getElementById('main-loading-bar');
// loading.setAttribute('title', url);
// if(!progress){
// progress = document.createElement('div');
```

```

// progress.setAttribute('class', 'loading-progress');
// loading.appendChild(progress);
// }
// progress.style.width = (((i - 1)/count) * 100) + '%';
// }
</script>
</head>

<body class="claro jimu-main-font">
<div id="main-loading">
<!-- This is section you can modify to customize the loading page --&gt;
&lt;div id="app-loading"&gt;&lt;/div&gt;
&lt;div id="loading-gif"&gt;&lt;/div&gt;
&lt;!-- ////////////////////// END ////////////////////// --&gt;
&lt;!--[if IE 8]&gt;
&lt;div id="ie-note" style="display:none;"&gt;
&lt;div class="hint-title"&gt;Error&lt;/div&gt;
&lt;div class="hint-img"&gt;Your browser is currently not supported.&lt;/div&gt;
&lt;p class="hint-text"&gt;
&lt;span&gt;
Please note that creating presentations is not supported in Internet
Explorer versions 6, 7.
&lt;/span&gt;
&lt;br&gt;
&lt;span&gt;
We recommend upgrading to the latest Internet Explorer, Google
Chrome, or Firefox.
&lt;/span&gt;
&lt;br&gt;
&lt;span&gt;
If you are using IE 8 or later, make sure you turn off "Compatibility View".
&lt;/span&gt;
&lt;/p&gt;
&lt;/div&gt;
&lt;![endif]--&gt;
&lt;/div&gt;
&lt;div id="main-page" role="application"&gt;
&lt;div class="trapLinkNode" tabindex="0"&gt;&lt;/div&gt;
&lt;div id="jimu-layout-manager"&gt;&lt;/div&gt;
&lt;div id="skipContainer"&gt;&lt;/div&gt;
&lt;div class="trapLinkNode lastTrapLinkNode" tabindex="0"&gt;&lt;/div&gt;
&lt;/div&gt;
&lt;div id="unsupported-browser-message"&gt;
&lt;div class="browser-message"&gt;
</pre>

```

```

<div class="browser-button-container">
  <button></button>
</div>
<h1 class="browser-message-title" id="message-title"></h1>
<p class="browser-message-content" id="message-content-1"></p>
<p class="browser-message-content" id="message-content-2"></p>
<p class="browser-message-content" id="message-content-3"></p>
<ul id="browser-logo-list">
  <li class="browser-logo"><a href="https://www.google.com/chrome/"></a></li>
  <li class="browser-logo"><a href="https://www.mozilla.org/firefox/"></a></li>
  <li class="browser-logo"><a href="https://www.apple.com/safari/"></a></li>
  <li class="browser-logo"><a href="https://www.microsoft.com/edge"></a></li>
</ul>
</div>
</div>

<script type="text/javascript" src=".//service-worker-registration.js"></script>
<script type="text/javascript" src="env.js"></script>
<script type="text/javascript" src="browser-message/limited-unsupported-browser.js"></script>
<script type="text/javascript" src="simpleLoader.js"></script>
<script type="text/javascript" src="init.js"></script>
</body>
</html>

```

Global variables:

- isXT: XT builder or not. This variable is set by builder node server.
- isBuilder: is builder or app. This flag is used to distinguish the build and app in jimu.js
- isRunInPortal: whether the app/builder is in portal or not.
- builderNls: the builder nls bundle
- portalUrl: the portal url
- configWindow: in builder only, the config app window object
- previewWindow: in builder only, the preview app window object
- setConfigViewerTopic: in builder only, used to communicate between builder and config window
- setPreviewViewerTopic: in builder only, used to communicate between builder and preview window
- appId: in builder only, the current opened app id.
- apiUrl: the URL of the ArcGIS API for JavaScript
- allCookies: all cookies

path: the builder/app path section in window.location
 appPath: the app's path. In app, it's the same with path; In builder, it's not.
 queryObject: the query parameters
 jimuNls: the jimu nls bundle
 isRTL: the language is right to left,
 wabVersion: the wab version, like 1.1, 1.2
 productVersion: the product version, like portal10.3, online3.5, developer edition1.0

dojoConfig: the dojo config object
 jimuConfig: defined in jimu/main

weinreUrl: for mobile debug
 debug: boolean. If it's debug mode, the app will load weinre file

Global functions:

```

loadResource: load JS or CSS
loadResources: load array of JS or CSS
testLoad: load JS/CSS by condition
loadingCallback: the resources loaded callback
******/  

/*global testLoad, ActiveXObject */  

var
  // the URL of the ArcGIS API for JavaScript, you can change it to point to your own API.
  apiUrl = null,  
  

  //weinreUrl: String
  // weinre is a tool which can help debug the app on mobile devices.
  // Please see: http://people.apache.org/~pmuellr/weinre/docs/latest/Home.html
  weinreUrl = '//launch.chn.esri.com:8081/target/target-script-min.js',  
  

  //debug: Boolean
  // If it's debug mode, the app will load weinre file
  debug = false,  
  

  //deprecated, use appInfo.appPath instead
  path = null,  
  

  isXT = false,  
  

  allCookies,  
  

  verboseLog = true,  
  

  //This version number will be appended to URL to avoid cache.
  //The reason we do not use wabVersion is to avoid force user to change wabVersion when they
  are customizing app.
  deployVersion = '2.26';  
  

  // console.time('before map');  
  

  ////Builder server will remove this line's comment, so set isXT flag to true
  //isXT = true;

```

```

//////////Build scripts will uncomment this line to disable verboseLog.

//verboseLog = false;

//////////

(function(global){
  //init API URL
  var queryObject = getQueryObject();
  var apiVersion = '3.42';

  ////////uncomment the following line when downloading the app

  //apiUrl = 'https://js.arcgis.com/3.42';

  ///////////
  allCookies = getAllCookies();

  if (queryObject.apiurl) {
    if(!checkApiUrl(queryObject.apiurl)){
      console.error('?apiurl must point to an ULR that is in the app or in esri.com/arcgis.com
domain.');
      return;
    }
    apiUrl = queryObject.apiurl;
  }
  window.appInfo = {isRunInPortal: !isXT};
  if (!apiUrl) {
    if (isXT) {
      apiUrl = 'https://js.arcgis.com/' + apiVersion;
    } else {
      var portalUrl = getPortalUrlFromLocation();
      if (portalUrl.indexOf('.arcgis.com') > -1) {
        if(portalUrl.indexOf('devext.arcgis.com') > -1){
          apiUrl = '//jsdev.arcgis.com/' + apiVersion;
        }else if(portalUrl.indexOf('qa.arcgis.com') > -1){
          apiUrl = '//jsqa.arcgis.com/' + apiVersion;
        }else{
          apiUrl = '//js.arcgis.com/' + apiVersion;
        }
      }
      // apiUrl = 'https://js.arcgis.com/' + apiVersion;
    } else {
      apiUrl = portalUrl + 'jsapi/jsapi/';
    }
  }
}

if (apiUrl.substr(apiUrl.length - 1, apiUrl.length) !== '/') {
  apiUrl = apiUrl + '/';
}

```

```

path = getPath();

function getAllCookies(){
    var strAllCookie = document.cookie;
    var cookies = {};
    if (strAllCookie) {
        var strCookies = strAllCookie.split('\'');
        for(var i = 0; i < strCookies.length; i++){
            var splits = strCookies[i].split('=');
            if(splits && splits.length > 1){
                cookies[splits[0].replace(/\s+/g, '')] = splits[1];
            }
        }
    }
    return cookies;
}

function checkApiUrl(url){
    if(/^\/\/.test(url) || ^https?:\/\/.test(url)){
        return /(?:[\w\-\_]+\.)+(?:esri|arcgis)\.com/.test(url); //api url must be in esri.com or
arcgis.com
    }else{
        return true;
    }
}

function getPortalUrlFromLocation(){
    var portalUrl = getPortalServerFromLocation() + getDeployContextFromLocation();
    return portalUrl;
}

function getPortalServerFromLocation(){
    var server = window.location.protocol + '/' + window.location.host;
    return server;
}

function getDeployContextFromLocation (){
    var keyIndex = window.location.href.indexOf("/home/");
    if(keyIndex < 0){
        keyIndex = window.location.href.indexOf("/apps/");
    }
    var context = window.location.href.substring(window.location.href.indexOf(
        window.location.host) + window.location.host.length + 1, keyIndex);
    if (context !== "/") {
        context = "/" + context + "/";
    }
    return context;
}

function getPath() {
    var fullPath, path;

    fullPath = window.location.pathname;

```

```

if (fullPath === '/' || fullPath.substr(fullPath.length - 1) === '/') {
  path = fullPath;
} else{
  var sections = fullPath.split('/');
  var lastSection = sections.pop();
  if (/\.html$/.test(lastSection) || /\.aspx$/.test(lastSection) ||
    /\.jsp$/.test(lastSection) || /\.php$/.test(lastSection)) {
    //index.html may be renamed to index.jsp, etc.
    path = sections.join('/') + '/';
  } else {
    return false;
  }
}
return path;
}

function getQueryObject(){
  var query = window.location.search;
  if (query.indexOf('?') > -1) {
    query = query.substr(1);
  }
  var pairs = query.split('&');
  var queryObject = {};
  for(var i = 0; i < pairs.length; i++){
    var splits = decodeURIComponent(pairs[i]).split('=');
    queryObject[splits[0]] = splits[1];
  }
  return queryObject;
}
function _loadPolyfills(prePath, cb) {
  prePath = prePath || "";
  var ap = Array.prototype,
    fp = Function.prototype,
    sp = String.prototype,
    loaded = 0,
    completeCb = function() {
      loaded++;
      if (loaded === tests.length) {
        cb();
      }
    },
    tests = [
      {
        test: window.console,
        failure: prePath + "libs/polyfills/console.js",
        callback: completeCb
      },
      {
        test: ap.indexOf && ap.lastIndexOf && ap.forEach && ap.every && ap.some &&
          ap.filter && ap.map && ap.reduce && ap.reduceRight,
        failure: prePath + "libs/polyfills/array.generics.js",
        callback: completeCb
      },
      {
        test: fp.bind,
        failure: prePath + "libs/polyfills/bind.js",
        callback: completeCb
      }
    ];
}

```

```

    }, {
      test: Date.now,
      failure: prePath + "libs/polyfills/now.js",
      callback: completeCb
    }, {
      test: sp.trim,
      failure: prePath + "libs/polyfills/trim.js",
      callback: completeCb
    }, {
      test: window.Blob,
      failure: prePath + "libs/polyfills/Blob.js",
      callback: completeCb
    }, {
      test: window.ArrayBuffer,
      failure: prePath + "libs/polyfills/typedarray.js",
      callback: completeCb
    }, {
      test: Object.assign,
      failure: prePath + "libs/polyfills/assign.js",
      callback: completeCb
    }, {
      test: Array.prototype.includes,
      failure: prePath + "libs/polyfills/array.includes.js",
      callback: completeCb
    }, {
      test: String.prototype.includes,
      failure: prePath + "libs/polyfills/string.includes.js",
      callback: completeCb
    });
  }

  for(var i = 0; i < tests.length; i++){
    testLoad(tests[i]);
  }
}

function localeIsSame(locale1, locale2){
  return locale1.split('-')[0] === locale2.split('-')[0];
}

function _setRTL(locale){
  var rtlLocales = ["ar", "he"];
  var dirNode = document.getElementsByTagName("html")[0];
  var isRTLLocale = false;
  for (var i = 0; i < rtlLocales.length; i++) {
    if (localeIsSame(rtlLocales[i], locale)) {
      isRTLLocale = true;
    }
  }

  dirNode.setAttribute("lang", locale);
  if (isRTLLocale) {
    dirNode.setAttribute("dir", "rtl");
    dirNode.className += " esriRtl jimu-rtl";
    dirNode.className += " " + locale + " "
  }
}

```

```

        (locale.indexOf("-") !== -1 ? locale.split("-")[0] : "");
    }else {
        dirNode.setAttribute("dir", "ltr");
        dirNode.className += " esriLtr jimu-ltr";
        dirNode.className += " " + locale + " " +
            (locale.indexOf("-") !== -1 ? locale.split("-")[0] : "");
    }

    window.isRTL = isRTLLocale;
}

global._loadPolyfills = _loadPolyfills;
global.queryObject = queryObject;
global._setRTL = _setRTL;

global.avoidRequireCache = function(require){
    var dojoInject = require.injectUrl;
    require.injectUrl = function(url, callback, owner){
        url = appendDeployVersion(url);
        dojoInject(url, callback, owner);
    };
};

global.avoidRequestCache = function (aspect, requestUtil){
    aspect.after(requestUtil, 'parseArgs', function(args){
        args.url = appendDeployVersion(args.url);
        return args;
    });
};

function appendDeployVersion(url){
    if(/^http(s)?://.test(url) || /^\\proxy\\.js/.test(url) || /^\\/\\.test(url)){
        return url;
    }
    if(url.indexOf('?') > -1){
        url = url + '&wab_dv=' + deployVersion;
    }else{
        url = url + '?wab_dv=' + deployVersion;
    }
    return url;
}

var _detectUserAgent = function() {
    var os = {}, browser = {},
        ua = navigator.userAgent, platform = navigator.platform,
        webkit = ua.match(/Web[kK]it[\\]{0,1}([\\d.]+)/),
        android = ua.match(/(Android);?[(\\s\\/]+([\\d.]+)?/),
        osx = !ua.match(/\Macintosh\\; Intel /),
        ipad = ua.match(/(iPad).*OS\\s([\\d.]+)/),
        ipod = ua.match(/(iPod).*(OS\\s([\\d.]+))?/),
        iphone = !ipad && ua.match(/(iPhone|sOS)\\s([\\d.]+)/),
        webos = ua.match(/(webOS|hpwOS)[\\s\\/](\\d.+)/),
        win = /Win\\d{2}|Windows/.test(platform),
        wp = ua.match(/Windows Phone ([\\d.]+)/),

```

```

touchpad = webos && ua.match(/TouchPad/),
kindle = ua.match(/Kindle\:\/\/([\d.]+)\/),
silk = ua.match(/Silk\:\/\/([\d._]+)\/),
blackberry = ua.match(/(BlackBerry).*Version\:\/\/([\d.]+)\/),
bb10 = ua.match(/(BB10).*Version\:\/\/([\d.]+)\/),
rimtabletos = ua.match(/(RIM\sTablet\sOS)\s([\d.]+)\/),
playbook = ua.match(/PlayBook/),
chrome = ua.match(/Chrome\:\/\/([\d.]+)\/ || ua.match(/CriOS\:\/\/([\d.]+)\/),
firefox = ua.match(/Firefox\:\/\/([\d.]+)\/),
firefoxos = ua.match(/\((?:Mobile|Tablet); rv:([\d.]+)\).*Firefox\:\/\/([\d.]+)\/),
ie = ua.match(/MSIE\s([\d.]+)\/ || ua.match(/Trident\:\/\/[\d](?=[^\?]+).*rv:([0-9.].+)\/),
webview = !chrome && ua.match(/(iPhone|iPod|iPad).*AppleWebKit(?!.*Safari)\/),
safari = webview || ua.match(/Version\:\/\/([\d.+])([^S](Safari)|[^M]*(Mobile)[^S]*(Safari))\/);

browser.webkit = !!webkit;
if (browser.webkit) {
    browser.version = webkit[1];
}

if (android) {
    os.android = true;
    os.version = android[2];
}
if (iphone && !ipod) {
    os.ios = os.iphone = true;
    os.version = iphone[2].replace(/_/g, '.');
}
if (ipad) {
    os.ios = os.ipad = true;
    os.version = ipad[2].replace(/_/g, '.');
}
if (ipod) {
    os.ios = os.ipod = true;
    os.version = ipod[3] ? ipod[3].replace(/_/g, '.') : null;
}
if (wp) {
    os.wp = true;
    os.version = wp[1];
}
if (webos) {
    os.webos = true;
    os.version = webos[2];
}
if (touchpad) {
    os.touchpad = true;
}
if (blackberry) {
    os.blackberry = true;
    os.version = blackberry[2];
}
if (bb10) {
    os.bb10 = true;
    os.version = bb10[2];
}

```

```

if (rimtabletos) {
    os.rimtabletos = true;
    os.version = rimtabletos[2];
}
if (playbook) {
    browser.playbook = true;
}
if (kindle) {
    os.kindle = true;
    os.version = kindle[1];
}
if (silk) {
    browser.silk = true;
    browser.version = silk[1];
}
if (!silk && os.android && ua.match(/Kindle Fire/)) {
    browser.silk = true;
}
if (chrome) {
    browser.chrome = true;
    browser.version = chrome[1];
}
if (firefox) {
    browser.firefox = true;
    browser.version = firefox[1];
}
if (firefoxos) {
    os.firefoxos = true;
    os.version = firefoxos[1];
}
if (ie) {
    browser.ie = true;
    browser.version = ie[1];
}
if (safari && (osx || os.ios || win)) {
    browser.safari = true;
    if (!os.ios) {
        browser.version = safari[1];
    }
}
if (webview) {
    browser.webview = true;
}

os.tablet = !(ipad || playbook || (android && !ua.match(/Mobile/)) ||
(firefox && ua.match(/Tablet/)) || (ie && !ua.match(/Phone/) && ua.match(/Touch/)));
os.phone = !(os.tablet && !os.ipod && (android || iphone || webos || blackberry || bb10 ||
(chrome && ua.match(/Android/)) || (chrome && ua.match(/CriOS\([^\d]+\)/)) ||
(firefox && ua.match(/Mobile/)) || (ie && ua.match(/Touch/))));

return {
    os: os,
    browser: browser
};

```

```

};

var _isMobileUa = function() {
    var uaInfo = global.userAgent;
    if (true === uaInfo.os.phone || true === uaInfo.os.tablet) {
        return true;
    } else {
        return false;
    }
};

global.userAgent = _detectUserAgent();
global.isMobileUa = _isMobileUa();
})(window);

window.showWarningForLimitedBrowser = function(jimuNls){
    var userAgent = window.navigator.userAgent || "";
    var unsupportedBrowser = /Trident\:\/\/d+/.test(userAgent); //IE

    var isInBuilder = isInConfigOrPreviewWindow();
    if(isInBuilder || !unsupportedBrowser){
        return;
    }

    var browserContainer = document.getElementById('unsupported-browser-message');
    var closeButton = document.getElementsByClassName('browser-button-container')[0].firstElementChild;
    //show message popup.
    browserContainer.style.display = 'flex';

    function isInConfigOrPreviewWindow(){
        var b = false;
        try{
            b = !window.isBuilder && window.parent && window.parent !== window &&
                window.parent.isBuilder;
        }catch(e){
            b = false;
        }
        return !!b;
    };

    function loadLocaleAndInitLinks(){
        var browserLabels = jimuNls.limitedUnsupportedBrowser;
        var closeLabel = jimuNls.common.close;
        closeButton.firstChild.setAttribute('alt', closeLabel);
        closeButton.firstChild.setAttribute('title', closeLabel);
        closeButton.firstChild.setAttribute('aria-label', closeLabel);

        document.getElementById('message-title').innerHTML = browserLabels.title;

        document.getElementById('message-content-1').innerHTML = browserLabels.onlineContent1;
        // document.getElementById('message-content-1').innerHTML = browserLabels.enterpriseContent1;

        document.getElementById('message-content-2').innerHTML = browserLabels.content2
    }
}

```

```

.replace(/<chrome\>-link>(.)</chrome\>-link>/, '<a class="browser-message-link"
href="https://www.google.com/chrome/">$1</a>')
.replace(/<firefox\>-link>(.)</firefox\>-link>/, '<a class="browser-message-link"
href="https://www.mozilla.org/firefox/">$1</a>')
.replace(/<safari\>-link>(.)</safari\>-link>/, '<a class="browser-message-link"
href="https://www.apple.com/safari/">$1</a>')
.replace(/<edge\>-link>(.)</edge\>-link>/, '<a class="browser-message-link"
href="https://www.microsoft.com/edge/">$1</a>');

document.getElementById('message-content-3').innerHTML = browserLabels.onlineContent3
.replace(/<feedback\>-link>(.)</feedback\>-link>/, '<a class="browser-message-link"
href="https://community.esri.com/community/gis/web-gis/arcgisonline">$1</a>');
}

function initEvents(){
  browserContainer.firstChild.onclick = function(evt){
    if(evt.target.tagName !== 'A' && evt.target.tagName !== 'IMG'){
      evt.preventDefault();
      evt.stopPropagation();
    }
  }
}

//close modal
browserContainer.onclick = function(){
  closeModal();
}
closeButton.onclick = function(){
  closeModal();
}
browserContainer.addEventListener('keydown', function(evt){
  if(evt.keyCode === 27){
    evt.preventDefault();
    closeModal();
  }
})

//for links inside modal
var links = document.getElementsByTagName('A');
var lastLink = links[links.length - 1];
closeButton.addEventListener('keydown', function(evt){
  if(evt.keyCode === 9 && evt.shiftKey){
    evt.preventDefault();
    lastLink.focus();
  }
})
lastLink.addEventListener('keydown', function(evt){
  if(evt.keyCode === 9 && !evt.shiftKey){
    evt.preventDefault();
    closeButton.focus();
  }
})
}

function closeModal(){

```

```

        browserContainer.style.display = 'none';
    }

    loadLocaleAndInitLinks();
    initEvents();
}

(function(global){

    //load js, css files
    function loadResources(ress, onOneBeginLoad, onOneLoad, onLoad){
        var loaded = [];
        function _onOneLoad(url){
            //to avoid trigger onload more then one time
            if(checkHaveLoaded(url)){
                return;
            }
            loaded.push(url);
            if(onOneLoad){
                onOneLoad(url, loaded.length);
            }
            if(loaded.length === ress.length){
                if(onLoad){
                    onLoad();
                }
            }
        }
    }

    for(var i = 0; i < ress.length; i ++){
        loadResource(ress[i], onOneBeginLoad, _onOneLoad);
    }

    function checkHaveLoaded(url){
        for(var i = 0; i < loaded.length; i ++){
            if(loaded[i] === url){
                return true;
            }
        }
        return false;
    }
}

function getExtension(url) {
    url = url || "";
    var items = url.split("?")[0].split(".");
    return items[items.length-1].toLowerCase();
}

function loadResource(url, onBeginLoad, onLoad){
    if(onBeginLoad){
        onBeginLoad(url);
    }
}

```

```

}

var type = getExtension(url);
if(type.toLowerCase() === 'css'){
  loadCss(url);
}else{
  loadJs(url);
}

function createElement(config) {
  var e = document.createElement(config.element);
  for (var i in config) {
    if (i !== 'element' && i !== 'appendTo') {
      e[i] = config[i];
    }
  }
  var root = document.getElementsByName(config.appendTo)[0];
  return (typeof root.appendChild(e) === 'object');
}

function loadCss(url) {
  var result = createElement({
    element: 'link',
    rel: 'stylesheet',
    type: 'text/css',
    href: url,
    onload: function(){
      elementLoaded(url);
    },
    appendTo: 'head'
  });

  //for the browser which doesn't fire load event
  //safari update documents.stylesheets when style is loaded.
  var ti = setInterval(function() {
    var styles = document.styleSheets;
    for(var i = 0; i < styles.length; i ++){
      // console.log(styles[i].href);
      if(styles[i].href &&
        styles[i].href.substr(styles[i].href.indexOf(url), styles[i].href.length) === url){
        clearInterval(ti);
        elementLoaded(url);
      }
    }
  }, 500);

  return (result);
}

function loadJs(url) {
  var result = createElement({
    element: 'script',
    type: 'text/javascript',
    onload: function(){
      elementLoaded(url);
    }
  });
}

```

```

        },
        onreadystatechange: function(){
            elementReadyStateChanged(url, this);
        },
        src: url,
        appendTo: 'body'
    });
    return (result);
}

function elementLoaded(url){
    if(onLoad){
        onLoad(url);
    }
}
function elementReadyStateChanged(url, thisObj){
    if (thisObj.readyState === 'loaded' || thisObj.readyState === 'complete') {
        elementLoaded(url);
    }
}
*******/

testLoad({
    test: window.console,
    success
})
*******/
function testLoad(testObj){
    testObj.success = !testObj.success? isArray(testObj.success)?
        testObj.success: [testObj.success]: [];
    testObj.failure = !!testObj.failure?
        isArray(testObj.failure)? testObj.failure: [testObj.failure]: [];

    if(testObj.test && testObj.success.length > 0){
        loadResources(testObj.success, null, null, testObj.callback);
    }else if(!testObj.test && testObj.failure.length > 0){
        loadResources(testObj.failure, null, null, testObj.callback);
    }else{
        testObj.callback();
    }
}

/* A must read: http://bonsaiden.github.com/JavaScript-Garden
 *****/
function is(type, obj) {
    var clas = Object.prototype.toString.call(obj).slice(8, -1);
    return obj !== undefined && obj !== null && clas === type;
}

function isArray(item) {
    return is("Array", item);
}

```

```

global.loadResources = loadResources;
global.loadResource = loadResource;
global.testLoad = testLoad;
}
)(window);

var dojoConfig, jimuConfig;

/*global weinreUrl, loadResources, _loadPolyfills, loadingCallback, debug, allCookies */

var ie = (function() {

    var undef,
        v = 3,
        div = document.createElement('div'),
        all = div.getElementsByTagName('i');

    div.innerHTML = '<!--[if gt IE ' + (++v) + ']><i></i><![endif]-->';
    while(all[0]){
        div.innerHTML = '<!--[if gt IE ' + (++v) + ']><i></i><![endif]-->';
    }
    return v > 4 ? v : undef;
}());

(function(argument) {
    if (ie < 8){
        var mainLoading = document.getElementById('main-loading');
        var appLoading = document.getElementById('app-loading');
        var ieNotes = document.getElementById('ie-note');
        appLoading.style.display = 'none';
        ieNotes.style.display = 'block';
        mainLoading.style.backgroundColor = "#fff";
        return;
    }

    //handle edit=true parameter
    if(!window.isXT && window.location.pathname.indexOf('/apps/webappviewer') > -1 &&
        window.queryObject.edit === 'true' && window.queryObject.appid){
        window.location.href = window.location.href.replace('webappviewer', 'webappbuilder');
        return;
    }

    var resources = [];
    if (debug) {
        resources.push(weinreUrl);
    }

    if (!window.apiUrl) {
        console.error('no apiUrl.');
    } else if (!window.path) {
        console.error('no path.');
    } else {

```

```

if(window.location.protocol === 'https:'){
  var reg = /^http:\/\/\//i;
  if(reg.test(window.apiUrl)){
    window.apiUrl = window.apiUrl.replace(reg, 'https://');
  }
  if(reg.test(window.path)){
    window.path = window.path.replace(reg, 'https://');
  }
}

/*jshint unused:false*/
var hasOptions = {'extend-esri': 1};
if(!window.isMobileUa && queryObject.disableFLWebGL !== '1' && queryObject.disableFLWebGL !== 'true'){
  hasOptions['esri-featurelayer-webgl'] = 1; //enable webgl
}
dojoConfig = {
  parseOnLoad: false,
  async: true,
  tlmSiblingOfDojo: false,
  has: hasOptions
};

setLocale();

if(window.isRTL){
  dojoConfig.has['dojo-bidi'] = true;
}

resources = resources.concat([
  window.apiUrl + 'dojo/resources/dojo.css',
  window.apiUrl + 'dijit/themes/claro/claro.css',
  window.apiUrl + 'esri/css/esri.css',
  // window.apiUrl + 'dojox/layout/resources/ResizeHandle.css',
  window.path + 'jimu.js/css/jimu-theme.css'
  //window.path + 'libs/caja-html-sanitizer-minified.js'
  //window.path + 'libs/moment/twix.js',
  //window.path + 'libs/Sortable.js',
  // window.path + 'libs/cropperjs/cropperjs.js',
  // window.path + 'libs/cropperjs/cropper.css',
  // //because we have jimu/dijit/GridLayout dijit, so we import this css here
  // window.path + 'libs/goldenlayout/goldenlayout-base.css',
  // window.path + 'libs/goldenlayout/goldenlayout-light-theme.css'
]);

if (window.apiUrl.substr(window.apiUrl.length - 'arcgis-js-api/'.length,
  'arcgis-js-api/'.length) === 'arcgis-js-api/') {
  //after build, we put js api here
  //user can also download release api package and put here
  dojoConfig.baseUrl = window.path;
  dojoConfig.packages = [
    {
      name: "dojo",
      location: window.apiUrl + "dojo"
    },
    {
      name: "esri",
      location: window.apiUrl + "esri"
    }
  ];
}

```

```

        name: "dijit",
        location: window.apiUrl + "dijit"
    },
    name: "dojox",
    location: window.apiUrl + "dojox"
},
{
    name: "put-selector",
    location: window.apiUrl + "put-selector"
},
{
    name: "xstyle",
    location: window.apiUrl + "xstyle"
},
{
    name: "dgrid",
    location: window.apiUrl + "dgrid"
},
{
    name: "dgrid1",
    location: window.apiUrl + "dgrid1"
},
{
    name: "dstore",
    location: window.apiUrl + "dstore"
},
{
    name: "moment",
    location: window.apiUrl + "moment"
},
{
    name: "esri",
    location: window.apiUrl + "esri"
},
{
    name: "widgets",
    location: "widgets"
},
{
    name: "jimu",
    location: "jimu.js"
},
{
    name: "themes",
    location: "themes"
},
{
    name: "libs",
    location: "libs"
},
{
    name: "dynamic-modules",
    location: "dynamic-modules"
}];

resources.push(window.apiUrl + '/dojo/dojo.js');
} else {
    dojoConfig.baseUrl = window.apiUrl + 'dojo';
    dojoConfig.packages = [
        {
            name: "widgets",
            location: window.path + "widgets"
        },
        {
            name: "jimu",
            location: window.path + "jimu.js"
        },
        {
            name: "themes",

```

```

        location: window.path + "themes"
    }, {
        name: "libs",
        location: window.path + "libs"
    }, {
        name: "dynamic-modules",
        location: window.path + "dynamic-modules"
    }, {
        name: "configs",
        location: window.path + "configs"
    });
}

resources.push(window.apiUrl + 'init.js');
}

jimuConfig = {
    loadingId: 'main-loading',
    mainPageId: 'main-page',
    layoutId: 'jimu-layout-manager',
    mapId: 'map'
};
jimuConfig.lazyLoadCss = [
    // 'xstyle/css!' + window.apiUrl + 'dojo/resources/dojo.css',
    // 'xstyle/css!' + window.apiUrl + 'dijit/themes/claro/claro.css',
    // 'xstyle/css!' + window.apiUrl + 'esri/css/esri.css',
    'xstyle/css!' + window.apiUrl + 'dojox/layout/resources/ResizeHandle.css'
    // 'xstyle/css!' + window.path + 'jimu.js/css/jimu-theme.css'
];
};

loadResources(resources, null, function(url, loaded) {
    if (typeof loadingCallback === 'function') {
        loadingCallback(url, loaded, resources.length);
    }
}, function() {
    continueLoad();
}

function continueLoad(){
    if(typeof require === 'undefined'){
        if (window.console){
            window.console.log('Waiting for API loaded.');
        }
        setTimeout(continueLoad, 50);
        return;
    }

    _loadPolyfills("", function() {
        window.appInfo.appPath = window.path;
        window.avoidRequireCache(require);
        require(['dojo/aspect', 'dojo/request/util'], function(aspect, requestUtil) {
            window.avoidRequestCache(aspect, requestUtil);

            require(['jimu/main'/*, 'libs/main'*/], function(jimuMain) {
                //loadingCallback('jimu', resources.length + 1, resources.length);
                window.showWarningForLimitedBrowser(window.jimuNls);
            });
        });
    });
}

```

```

        jimuMain.initApp();
    });
});
});
}
});
}

function setLocale(){
if(window.queryObject.locale){
    var locale = window.queryObject.locale.toLowerCase();
    dojoConfig.locale = ['hi'].indexOf(locale) >= 0 ? 'en' : locale;
    window._setRTL(dojoConfig.locale);
    return;
}

if(window.queryObject.mode){
    if(allCookies.wab_locale){
        dojoConfig.locale = allCookies.wab_locale;
    }
}else{
    if(allCookies.wab_app_locale){
        dojoConfig.locale = allCookies.wab_app_locale;
    }
}
}

if(!dojoConfig.locale){
    dojoConfig.locale = navigator.language ? navigator.language : navigator.userLanguage;
}

dojoConfig.locale = dojoConfig.locale.toLowerCase();
dojoConfig.locale = ['hi'].indexOf(dojoConfig.locale) >= 0 ? 'en' : dojoConfig.locale;
window._setRTL(dojoConfig.locale);
}

})();

define(["dojo/_base/declare",
        "dojo/_base/lang",
        "dojo/_base/array",
        "dojo/promise/all",
        "dojo/Deferred",
        "dojo/json",
        "dojo/i18n!widgets/AddToMap/nls/strings",
        "esri/lang",
        "esri/request",
        "esri/arcgis/utils",
        "esri/layers/ArcGISDynamicMapServiceLayer",
        "jimu/layer/JimuLayer"
    ],
    function(declare, lang, array, Promise, Deferred, json, strings, Lang, Request, Utils, ArcGISDynamicMapServiceLayer, JimuLayer) {
        var _this = this;
        ...
    }
);
}

```

```

    "esri/layers/ArcGISImageServiceLayer",
    "esri/layers/ArcGISTiledMapServiceLayer",
    "esri/layers/FeatureLayer",
    "esri/layers/ImageServiceParameters",
    "esri/layers/KMLLayer",
    "esri/layers/MosaicRule",
    "esri/layers/RasterFunction",
    "esri/layers/VectorTileLayer",
    'esri/layers/WMSLayer',
    "esri/dijit/PopupTemplate",
    "esri/InfoTemplate",
    "esri/renderers/jsonUtils"],

    function(declare, lang, array, all, Deferred, djJson, i18n, esriLang, esriRequest, agsUtils,
    ArcGISDynamicMapServiceLayer, ArcGISImageServiceLayer, ArcGISTiledMapServiceLayer,
    FeatureLayer,
    ImageServiceParameters, KMLLayer, MosaicRule, RasterFunction, VectorTileLayer, WMSLayer,
    PopupTemplate, InfoTemplate, jsonRendererUtils) {

        return declare(null, {

            item: null,
            itemUrl: null,
            map: null,
            serviceUrl: null,

            constructor: function(args) {
                lang.mixin(this,args);
            },

            addItem: function(item,map) {
                // TODO layer position, titles, timeouts, feedback
                //console.warn("addItem",item);
                var dfd = new Deferred();
                this.map = map;
                this.item = item;
                this.itemUrl = this._checkMixedContent(item.itemUrl);
            }
        });
    }
}

```

```

this.serviceUrl = this._checkMixedContent(item.url);

if (item.type === "Feature Service") {
    return this._addFeatureService();
} else if (item.type === "Image Service") {
    return this._addImageService();
} else if (item.type === "KML") {
    return this._addKML();
} else if (item.type === "Map Service") {
    return this._addMapService();
} else if (item.type === "Vector Tile Service") {
    return this._addVectorTileService();
} else if (item.type === "WMS") {
    return this._addWMS();
} else {
    // TODO reject
    console.warn("Unsupported item type: ", item.type);
    dfd.resolve(null);
}
return dfd;
},

_addFeatureService: function() {
    var self = this, dfd = new Deferred();
    var serviceUrl = this.serviceUrl;
    var item = this.item, itemData = {};
    var layerIds = null, layerDfds = [], featureLayers = [];

    self._readItemJsonData().then(function(result){
        //console.warn("_addFeatureService.jsonData", result);
        itemData = result || {};
        if (itemData && itemData.layers && (itemData.layers.length > 0)) {
            array.forEach(itemData.layers, function(l){
                if ((typeof(l.id) !== "undefined") && (l.id !== null)) {
                    if (layerIds === null) layerIds = [];
                    layerIds.push(l.id);
                }
            });
        }
    });
    dfd.resolve(serviceUrl);
    return dfd;
}

```

```

        }
    });
}

return self._readRestInfo(serviceUrl);

}).then(function(result){
//console.warn("_addFeatureService.serviceInfo",result);
if (result && typeof result.type === "string" && result.type === "Feature Layer") {
    // a single layer registered from a service /FeatureServer/1 or /MapServer/2
    var layer = new FeatureLayer(serviceUrl,{
        id: self._generateLayerId(),
        outFields: ["*"]
    });
    layerDfds.push(self._waitForLayer(layer));
} else if (result && result.layers && result.layers.length > 0) {
    array.forEach(result.layers,function(lyr){
        var bAdd = true;
        if (layerIds !== null && layerIds.length > 0) {
            bAdd = array.some(layerIds,function(lid){
                return (lid === lyr.id);
            });
        }
        if (bAdd) {
            var layer = new FeatureLayer(serviceUrl+"/"+lyr.id,{
                id: self._generateLayerId(),
                outFields: ["*"]
            });
            layerDfds.push(self._waitForLayer(layer));
        }
    });
}
return all(layerDfds);

}).then(function(results){
//console.warn("_addFeatureService.layerDfds",results);
array.forEach(results,function(result){

```

```

        featureLayers.push(result);
    });
    featureLayers.reverse();
    return featureLayers;
}

}).then(function(){
    array.forEach(featureLayers,function(layer){
        var opLayer = self._processFeatureLayer(layer,item,itemData);
        layer.arcgisProps = {title: opLayer.title};
        layer._titleForLegend = opLayer.title;
        if (!esriLang.isDefined(layer.title)) layer.title = opLayer.title;
        self._addLayer(layer);
    });
}).then(function(){
    dfd.resolve(featureLayers);
}).otherwise(function(error){
    dfd.reject(error);
});
return dfd;
},

```

```

_addImageService: function() {
    var self = this, dfd = new Deferred();
    self._readItemJsonData().then(function(result){
        var itemData = result || {};
        return self._newImageServiceLayer(itemData);
    }).then(function(layer){
        self._addLayer(layer);
        dfd.resolve(layer);
    }).otherwise(function(error){
        dfd.reject(error);
    });
    return dfd;
},

```

```

_addKML: function() {

```

```
var self = this, dfd = new Deferred();
self._newKMLLayer().then(function(layer){
    if (layer) layer.title = self.item.title;
    self._addLayer(layer);
    dfd.resolve(layer);
}).otherwise(function(error){
    dfd.reject(error);
});
return dfd;
},

_addMapService: function() {
    var self = this, dfd = new Deferred();
    self._readItemJsonData().then(function(result){
        var itemData = result || {};
        return self._newMapServiceLayer(itemData);
    }).then(function(layer){
        self._addLayer(layer);
        dfd.resolve(layer);
    }).otherwise(function(error){
        dfd.reject(error);
    });
    return dfd;
},

_addVectorTileService: function() {
    var self = this, dfd = new Deferred();
    self._newVectorTileLayer().then(function(layer){
        self._addLayer(layer);
        dfd.resolve(layer);
    }).otherwise(function(error){
        dfd.reject(error);
    });
    return dfd;
},
```

```

_addWMS: function() {
  var self = this, dfd = new Deferred();
  self._readItemJsonData().then(function(result){
    var itemData = result || {};
    return self._newWMSLayer(itemData);
  }).then(function(layer){
    if (layer) layer.title = self.item.title;
    self._addLayer(layer);
    dfd.resolve(layer);
  }).otherwise(function(error){
    dfd.reject(error);
  });
  return dfd;
},

```

```

_addLayer: function(layer) {
  //console.warn("_addLayer",layer);
  //console.warn("map",this.map);
  var item = this.item;
  if (layer) {
    layer.xtnItemId = item.id;
    if (!layer.arcgisProps && item) {
      layer.arcgisProps = {title: item.title};
      layer._titleForLegend = item.title;
    }
    if (!esriLang.isDefined(layer.title)) layer.title = item.title;
    this.map.addLayer(layer);
  }
},

```

```

_checkUrl: function(url) {
  return agsUtils._checkUrl(url);
},

```

```
_checkVectorTileUrl: function(url,operationalLayer) {
    var dfd = new Deferred();
    var endsWith = function(sv,sfx) {
        return (sv.indexOf(sfx,(sv.length - sfx.length)) !== -1);
    };
    if (endsWith(url,".json")) {
        operationalLayer.styleUrl = url;
        dfd.resolve(url);
        return dfd;
    }
    var params = {url:null,content:{},handleAs:"json",callbackParamName:"callback"};
    if (this.itemUrl) {
        params.url = this.itemUrl+"/resources/styles/root.json";
        esriRequest(params,{}).then(function(){
            operationalLayer.styleUrl = params.url;
            dfd.resolve(params.url);
        }).otherwise(function(){
            params.url = url+"/resources/styles/root.json";
            esriRequest(params,{}).then(function(){
                operationalLayer.styleUrl = params.url;
                dfd.resolve(params.url);
            }).otherwise(function(){
                operationalLayer.url = url;
                dfd.resolve(url);
            });
        });
    });
    } else {
        params.url = url+"/resources/styles/root.json";
        esriRequest(params,{}).then(function(){
            operationalLayer.styleUrl = params.url;
            dfd.resolve(params.url);
        }).otherwise(function(){
            operationalLayer.url = url;
            dfd.resolve(url);
        });
    }
}
```

```

        return dfd;
    },

    _generateLayerId: function() {
        return this._generateLayerIds(1)[0];
    },

    _generateLayerIds: function(count) {
        var i, ids = [];
        for (i=0;i<count;i++) {
            ids.push(this._generateRandomId());
        }
        return ids;
    },

    _generateRandomId: function() {
        var t = null;
        if (typeof Date.now === "function") t = Date.now();
        else t = (new Date()).getTime();
        var r = (""+Math.random()).replace("0.","r");
        return (t+"_"+r).replace(/-/g,"");
    },

    _makeFeatureLayerTitle: function(pattern,serviceName,layerName) {
        var n,s,regexp;
        try {
            if (serviceName && layerName && (serviceName === layerName)) {
                return serviceName;
            } else if (serviceName && layerName) {
                // try to remove a timestamp suffix
                n = layerName.indexOf(serviceName);
                if (n === 0) {
                    s = layerName.substring(n+serviceName.length+1);
                    if (s.length >= 13) {
                        regexp = /\^\\d+\$/;
                        if (regexp.test(s)) {

```

```

        return serviceName;
    }
}
}
}
}

} catch(ex) {}

return pattern.replace("{serviceName}",serviceName).replace("{layerName}",layerName);
},

_newImageServiceLayer: function(itemData) {
//console.warn("_newImageServiceLayer.itemData",itemData);

var dfd = new Deferred();
var mapLayerId = this._generateLayerId();
var layerUrl = this.serviceUrl;
var layerObject = {
    mapLayerId: mapLayerId,
    bandIds: null,
    format: null,
    compressionQuality: null,
    opacity: 1.0,
    visibility: true
};

if (esriLang.isDefined(itemData.visibility) && itemData.visibility === false) {
    layerObject.visibility = false; // TODO?
}

if (esriLang.isDefined(itemData.opacity)) {
    layerObject.opacity = itemData.opacity;
}

if (esriLang.isDefined(itemData.minScale) && !esriLang.isDefined(layerObject.minScale)) {
    layerObject.minScale = itemData.minScale;
}

if (esriLang.isDefined(itemData.maxScale) && !esriLang.isDefined(layerObject.maxScale)) {
    layerObject.maxScale = itemData.maxScale;
}
}

```

```

if (esriLang.isDefined(itemData.refreshInterval))
    && !esriLang.isDefined(layerObject.refreshInterval)) {
    layerObject.refreshInterval = itemData.refreshInterval;
}

if (itemData.popupInfo && !layerObject.popupInfo && !layerObject.disablePopup) {
    layerObject.popupInfo = itemData.popupInfo;
}

if (itemData.renderingRule && !layerObject.renderingRule) {
    layerObject.renderingRule = itemData.renderingRule;
    if (itemData.renderingRule.functionName) {
        layerObject.renderingRule.rasterFunction = itemData.renderingRule.functionName;
    }
}

if (itemData.bandIds && !layerObject.bandIds) {
    layerObject.bandIds = itemData.bandIds;
}

if (itemData.mosaicRule && !layerObject.mosaicRule) {
    layerObject.mosaicRule = itemData.mosaicRule;
}

if (itemData.format && !layerObject.format) {
    layerObject.format = itemData.format;
}

if (esriLang.isDefined(itemData.compressionQuality)
    && !esriLang.isDefined(layerObject.compressionQuality)) {
    layerObject.compressionQuality = itemData.compressionQuality;
}

if (itemData.layerDefinition && itemData.layerDefinition.definitionExpression &&
    (!esriLang.isDefined(layerObject.layerDefinition)
     || !esriLang.isDefined(layerObject.layerDefinition.definitionExpression))) {
    layerObject.layerDefinition = layerObject.layerDefinition || {};
    layerObject.layerDefinition.definitionExpression
    itemData.layerDefinition.definitionExpression;
}

var imageServiceParameters = new ImageServiceParameters();
//imageServiceParameters.bandIds = layerObject.bandIds;

```

```
if (layerObject.bandIds !== null) {
    imageServiceParameters.bandIds = layerObject.bandIds;
}

if (layerObject.format !== null) {
    imageServiceParameters.format = layerObject.format;
}

if (layerObject.compressionQuality !== null) {
    imageServiceParameters.compressionQuality = layerObject.compressionQuality;
}

}

if (layerObject.renderingRule && layerObject.renderingRule.rasterFunction) {
    var rasterFunction = new RasterFunction(layerObject.renderingRule);
    imageServiceParameters.renderingRule = rasterFunction;
}

if (layerObject.mosaicRule) {
    var mosaicRule = new MosaicRule(layerObject.mosaicRule);
    imageServiceParameters.mosaicRule = mosaicRule;
}

if (esriLang.isDefined(layerObject.noData)) {
    imageServiceParameters.noData = layerObject.noData;
}

if (esriLang.isDefined(layerObject.noDataInterpretation)) {
    imageServiceParameters.noDataInterpretation = layerObject.noDataInterpretation;
}

if (esriLang.isDefined(layerObject.interpolation)) {
    imageServiceParameters.interpolation = layerObject.interpolation;
}

var props = {
    imageServiceParameters: imageServiceParameters,
    opacity: layerObject.opacity,
    visible: layerObject.visibility
};

if (esriLang.isDefined(layerObject.mapLayerId)) {
    props.id = layerObject.mapLayerId;
}

if (esriLang.isDefined(layerObject.minScale)) {
```

```

    props.minScale = layerObject.minScale;
}

if (esriLang.isDefined(layerObject.maxScale)) {
    props.maxScale = layerObject.maxScale;
}

if (esriLang.isDefined(layerObject.refreshInterval)) {
    props.refreshInterval = layerObject.refreshInterval;
}

if (esriLang.isDefined(layerObject.resourceInfo)) {
    props.resourceInfo = layerObject.resourceInfo;
}

var finish = function(layer) {
    //console.warn("finish",layer);
    if (layerObject.layerDefinition && layerObject.layerDefinition.definitionExpression) {
        layer.setDefinitionExpression(layerObject.layerDefinition.definitionExpression,true);
    }
    // TODO setInfoTemplate
    //if (!options.ignorePopups && layerObject.popupInfo) {
    //    layer.setInfoTemplate(new clazz(layerObject.popupInfo));
    //}
    /*
    rasterUtil.populateLayerWROInfo(layer,true).then(
        function(){dfd.resolve(layer);},
        function(error2){dfd.reject(error2);}
    );
    */
    dfd.resolve(layer);
};

var lyr = new ArcGISImageServiceLayer(this._checkUrl(layerUrl),props);
this._waitForLayer(lyr).then(
    function(layer){finish(layer);},
    function(error){dfd.reject(error);}
);

```

```

        return dfd;
    },

    _newInfoTemplate: function(popupInfo,title) {
        if (popupInfo) {
            try {
                var popupTemplate = new PopupTemplate({
                    description: popupInfo.description,
                    title: popupInfo.title,
                    showAttachments: popupInfo.showAttachments,
                    fieldInfos: popupInfo.fieldInfos,
                    mediaInfos: popupInfo.mediaInfos
                });
                return popupTemplate;
            } catch(ex) {
                console.error(ex);
            }
        }
        var infoTemplate = new InfoTemplate();
        if (esriLang.isDefined(title)) infoTemplate.setTitle(title);
        return infoTemplate;
    },
}

_newKMLLayer: function() {
    var options = {id: this._generateLayerId()};
    var lyr = new KMLLayer(this.serviceUrl,options);
    return this._waitForLayer(lyr);
},
}

_newMapServiceLayer: function(itemData) {
    var self = this, dfd = new Deferred();
    var serviceUrl = this.serviceUrl;
    var mapLayerId = this._generateLayerId();
    var content = {f: "json"};
    esriRequest({url:serviceUrl,content:content,handleAs:"json",callbackParamName:"callback"},{}).then(

```

```

function(response) {
  var lyr = null;
  var options = {id: mapLayerId};
  if (response.tileInfo) {
    lyr = new ArcGISTiledMapServiceLayer(serviceUrl,options);
  } else {
    lyr = new ArcGISDynamicMapServiceLayer(serviceUrl,options);
  }
  self._waitForLayer(lyr).then(
    function(layer) {
      //console.warn("MapServiceLayer",layer);
      var templates = null;
      array.forEach(layer.layerInfos,function(layerInfo){
        //console.warn("MapServiceLayer.layerInfo",layerInfo);
        var cfgLyr = null;
        if (itemData) {
          array.some(itemData.layers,function(l){
            if (layerInfo.id === l.id) {
              cfgLyr = l;
              return true;
            }
          });
        }
        var popupInfo = null;
        if (cfgLyr && cfgLyr.popupInfo) popupInfo = cfgLyr.popupInfo;
        if (templates === null) templates = {};
        templates[layerInfo.id] = {
          infoTemplate: self._newInfoTemplate(popupInfo,layerInfo.name)
        };
      });
      if (layer.infoTemplates === null) {
        if (templates) layer.infoTemplates = templates;
      }
      dfd.resolve(layer);
    },
    function(error2) {dfd.reject(error2);}
  );
}

```

```

    );
},
function(error){dfd.reject(error);}
);
return dfd;
},
_newVectorTileLayer: function() {
var self = this, dfd = new Deferred(), opLayer = {};
var serviceUrl = this.serviceUrl;
var mapLayerId = this._generateLayerId();
if ((typeof serviceUrl === "string") && (serviceUrl.length > 0)) {
this._checkVectorTileUrl(serviceUrl,opLayer).then(
function(url){
if ((typeof url === "string") && (url.length > 0)) {
url = self._checkMixedContent(url);
var props = {
id: mapLayerId,
opacity: 1,
visible: true
};
//console.warn("url",url,props);
var lyr = new VectorTileLayer(url,props);
//console.warn("lyr",lyr);
self._waitForLayer(lyr).then(
function(layer){dfd.resolve(layer);},
function(error2){dfd.reject(error2);}
);
} else {
dfd.resolve(null);
}
},
function(error) {dfd.reject(error);}
);
} else {
dfd.resolve(null);
}
},
function(error) {dfd.reject(error);}
);
} else {
dfd.resolve(null);
}
}

```

```

    }

    return dfd;
},


_newWMSLayer: function() {
    var options = {id: this._generateLayerId()};
    var lyr = new WMSLayer(this.serviceUrl,options);
    var self = this, dfd = this._waitForLayer(lyr);
    dfd.then(function(layer){
        self._setWMSVisibleLayers(layer);
    });
    return dfd;
},


_checkMixedContent: function(uri) {
    if ((typeof window.location.href === "string") && (window.location.href.indexOf("https://") === 0)) {
        if ((typeof uri === "string") && (uri.indexOf("http://") === 0)) {
            uri = "https:"+uri.substring("5");
        }
    }
    return uri;
},


_processFeatureLayer: function(featureLayer,item,itemData) {
    var self = this, dlPattern = i18n.search.featureLayerTitlePattern;
    var opLayer = null;
    if (itemData && itemData.layers && (itemData.layers.length > 0)) {
        array.some(itemData.layers,function(info) {
            var layerDefinition, jsonRenderer, renderer, isCustomTemplate = false;
            var popInfo, jsonPopInfo, infoTemplate;
            if (info.id === featureLayer.layerId) {
                //console.warn("layerInfo",info);
                if (info.popupInfo) {
                    popInfo = info.popupInfo;
                    jsonPopInfo = djJson.parse(djJson.stringify(popInfo));

```

```
infoTemplate = new PopupTemplate(jsonPopInfo);
featureLayer.setInfoTemplate(infoTemplate);
isCustomTemplate = true;
}

if (esriLang.isDefined(info.showLabels)) {
    featureLayer.setShowLabels(info.showLabels);
}

if (esriLang.isDefined(info.refreshInterval)) {
    featureLayer.setRefreshInterval(info.refreshInterval);
}

if (esriLang.isDefined(info.showLegend)) {
    // TODO?
}

if (esriLang.isDefined(info.timeAnimation)) {
    if (info.timeAnimation === false) {
        // TODO?
    }
}

layerDefinition = info.layerDefinition;
if (layerDefinition) {
    if (layerDefinition.definitionExpression) {
        featureLayer.setDefinitionExpression(layerDefinition.definitionExpression);
    }

    if (layerDefinition.displayField) {
        featureLayer.displayField(layerDefinition.displayField);
    }

    if (layerDefinition.drawingInfo) {
        if (layerDefinition.drawingInfo.renderer) {
            jsonRenderer = djJson.parse(djJson.stringify(layerDefinition.drawingInfo.renderer));
            renderer = jsonRendererUtils.fromJson(jsonRenderer);
            if (jsonRenderer.type && (jsonRenderer.type === "classBreaks")) {
                renderer.isMaxInclusive = true;
            }
            featureLayer.setRenderer(renderer);
        }
    }

    if (esriLang.isDefined(layerDefinition.drawingInfo.transparency)) {
```

```

        // TODO validate before setting?
        featureLayer.setOpacity(1 - (layerDefinition.drawingInfo.transparency / 100));
    }
}

if (esriLang.isDefined(layerDefinition.minScale)) {
    featureLayer.setMinScale(layerDefinition.minScale);
}

if (esriLang.isDefined(layerDefinition.maxScale)) {
    featureLayer.setMaxScale(layerDefinition.maxScale);
}

if (esriLang.isDefined(layerDefinition.defaultVisibility)) {
    if (layerDefinition.defaultVisibility === false) {
        featureLayer.setVisibility(false); // TODO?
    }
}
}

if (!isCustomTemplate) {
    self._setFeatureLayerInfoTemplate(featureLayer,info.popupInfo);
}

opLayer = {
    url: featureLayer.url,
    id: featureLayer.id,
    itemId: item.id,
    title: self._makeFeatureLayerTitle(dlPattern,item.title,featureLayer.name)
};

return true;
}

});

return opLayer;

}

} else {
    opLayer = {
        url: featureLayer.url,
        id: featureLayer.id,
        itemId: item.id,
        title: self._makeFeatureLayerTitle(dlPattern,item.title,featureLayer.name)
    }
}
}
```

```

};

self._setFeatureLayerInfoTemplate(featureLayer,null,opLayer.title);

return opLayer;

}

},


_readItemJsonData: function() {

var u = this.itemUrl+"/data";

var content = {f:"json"}, options = {};

return esriRequest({url:u,content:content,handleAs:"json"},options);

},


_readRestInfo: function(url) {

return

esriRequest({url:url,content:{f:"json"},handleAs:"json",callbackParamName:"callback"},{});

},


_setFeatureLayerInfoTemplate: function(featureLayer,popupInfo,title) {

var template = this._newInfoTemplate(popupInfo,title);

featureLayer.setInfoTemplate(template);

},


_setWMSVisibleLayers: function(layer) {

var maxLayers = 10, lyrNames = [];

if (layer) {

array.some(layer.layerInfos,function(lyrInfo){

//console.warn("lyrInfo",lyrInfo);

if (typeof lyrInfo.name === "string" && lyrInfo.name.length > 0) {

if (lyrNames.length < maxLayers) {

lyrNames.push(lyrInfo.name);

} else {

return true;

}

}

});

//console.warn("lyrNames",lyrNames);
}
}
```

```

if (lyrNames.length <= maxLayers) {
    layer.setVisibleLayers(lyrNames);
}
},
};

_waitForLayer: function(layer) {
    var dfd = new Deferred(), handles = [];
    if (layer.loaded) {
        dfd.resolve(layer);
        return dfd;
    }
    if (layer.loadError) {
        dfd.reject(layer.loadError);
        return dfd;
    }
    var clearHandles = function() {
        array.forEach(handles,function(h){h.remove()});
    };
    //console.warn("_waitForLayer");
    handles.push(layer.on("load",function(layerLoaded){
        //console.warn("_waitForLayer.load",layerLoaded);
        clearHandles();
        dfd.resolve(layerLoaded.layer);
    }));
    handles.push(layer.on("error",function(layerError){
        //console.warn("_waitForLayer.error",layerError);
        clearHandles();
        var error = layerError.error;
        try {
            if (error.message && (error.message.indexOf("Unable to complete") !== -1)) {
                console.warn("layerAccessError",error);
                dfd.reject(new Error(i18n.search.layerInaccessible));
            } else {
                dfd.reject(error);
            }
        }
    }));
}
};

```

```
    } catch (ex) {
        //console.warn("layerAccessError",ex);
        dfd.reject(error);
    }
});  
return dfd;  
}  
  
});  
});
```