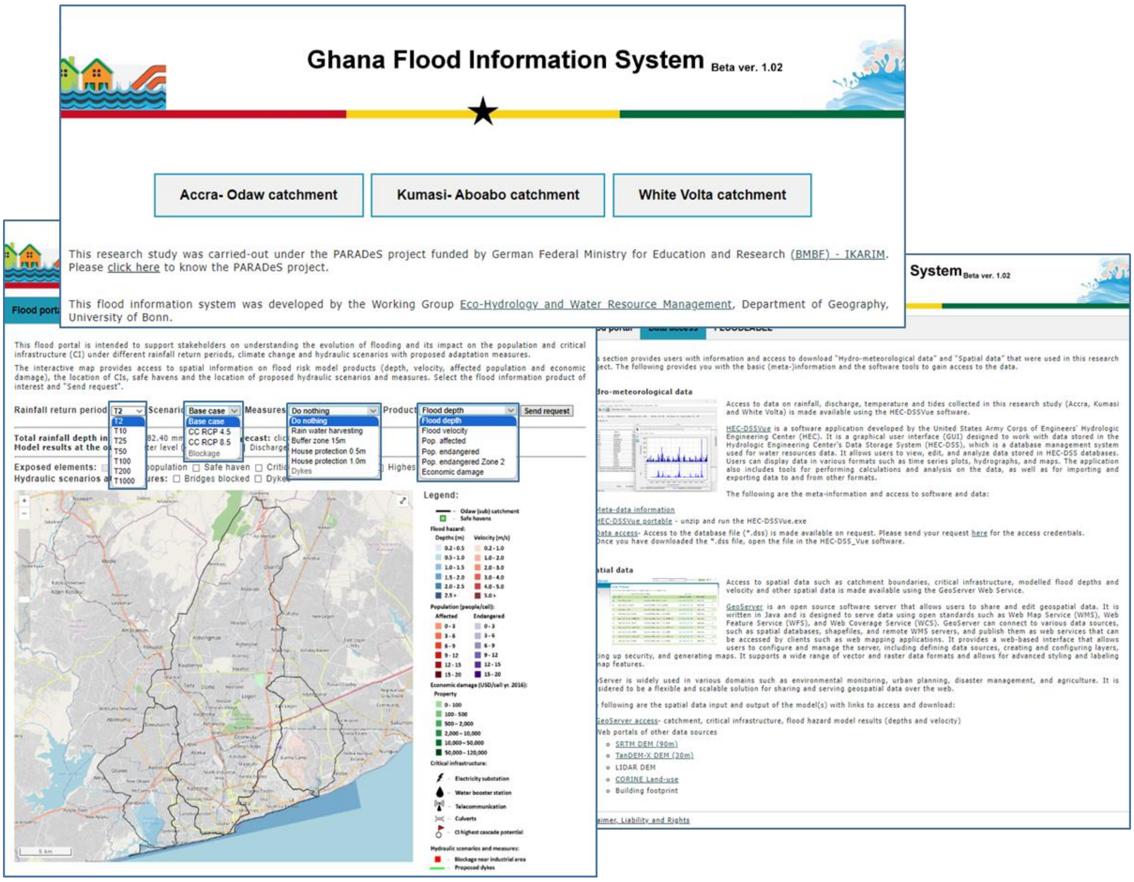
Ghana- Flood Information System (FIS)

Technical Documentation

PARADeS – Open Education Resource



Prepared by

Adrian Almoradie, PhD

Department of Geography, University of Bonn

September 2023

This document intends to furnish a comprehensive understanding of the technical aspects of the Flood Information System (FIS). This will encompass hardware, software and knowledge prerequisites, software setup and installation, as well as the creation of the web-GIS application and the publication of model results.

Table of Content

1. The Ghana-Flood Information System (FIS)

The Ghana-FIS developed for the catchments Odaw (Accra), Aboabo (Kumasi) and the White Volta is a web-based application to support and improve decision-making capabilities for Flood Disaster Risk Management (FDRM) by bringing together scientific data and local knowledge. It primarily serves as an informational tool rather than an operational application. The FIS comprises three key elements: the knowledge database, software, and a user interface that employs open-source software to simplify long-term maintenance, extending its usability beyond the project's duration.

In the following, we will illustrate the FIS components and their functions that would guide experts, researchers, and other stakeholders to access pertinent spatial information and data. This tool consolidates all relevant model outcomes into a user-friendly application with its core data repository that houses all the essential data gathered and generated through the PARADeS research project.

1.1 Flood Portal component

The intent behind this Flood portal (Fig. 1.) is to aid stakeholders in gaining a comprehensive understanding of how floods evolve and impact the population, economy and critical infrastructure (CI). This understanding is crucial under various conditions such as different rainfall return periods, climate change scenarios, and hydraulic situations, all while proposing potential adaptation measures.

Using the interactive map, users can access spatial information regarding flood hazard model outcomes, encompassing data on depth, velocity, and its consequence to the population and property damage. Additionally, the map highlights the geographical placement of critical infrastructure (CI), and safe havens, as well as the specific locations designated for hydraulic scenarios and mitigation measures. In more detail regarding the selection of flood products, users will have options to choose the return periods (T2 to T1000), scenarios of climate, measures if implemented and the type of product to present in the map. You can select your preferred flood information product and proceed to "Send a Request."

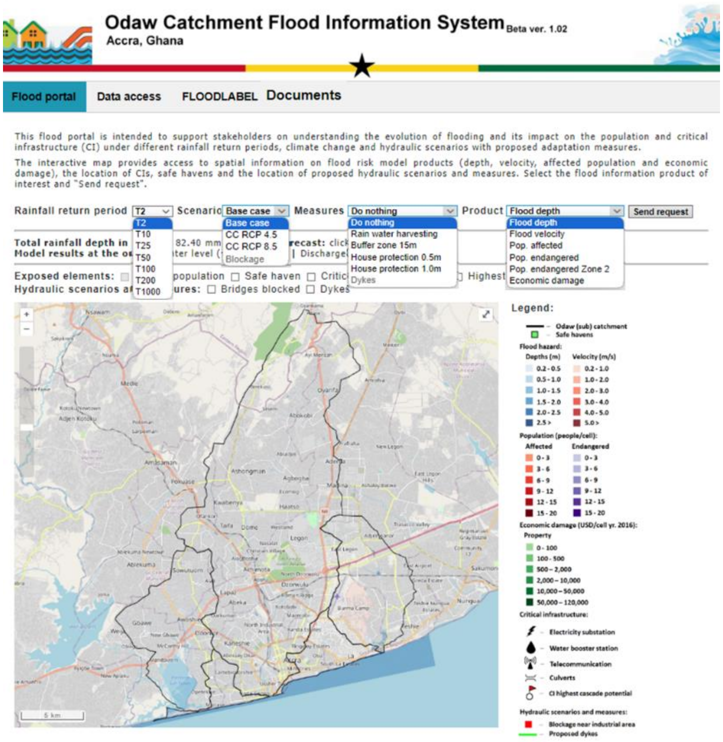


Figure 1: Flood portal component

1.2 Data Access component

This portal component (Fig. 2) provides users with meta-data information, access to download "*Hydro-meteorological data*" and "*Spatial data*" and the software tools to gain access to the data that were used in this research project.

Hydro-meteorological data

Access to data on rainfall, discharge, temperature and tides collected in this research study (Accra, Kumasi and White Volta) is made available using the HEC-DSSVue software.

Spatial data

Access to spatial data such as catchment boundaries, critical infrastructure, modelled flood depths and velocity and other spatial data is made available using the GeoServer Web Service.

Links to web portals containing input data from other sources are also shared.

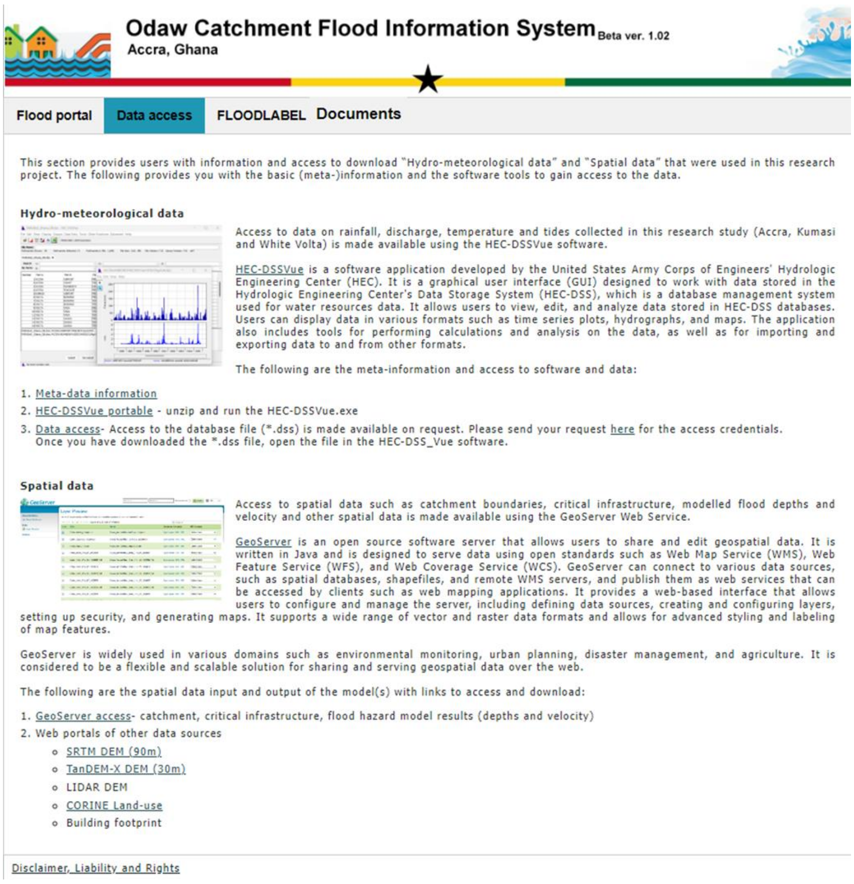


Figure 2: Data access component

1.3 FLOODLABEL component

In this component (Fig. 3), the FLOODLABEL concept and its concise explanation of how it can be utilized for property assessment and the recommendation of appropriate measures is presented. Furthermore, we offer a brief video and downloadable resources, including flyers, a measure booklet, and emergency response plans.

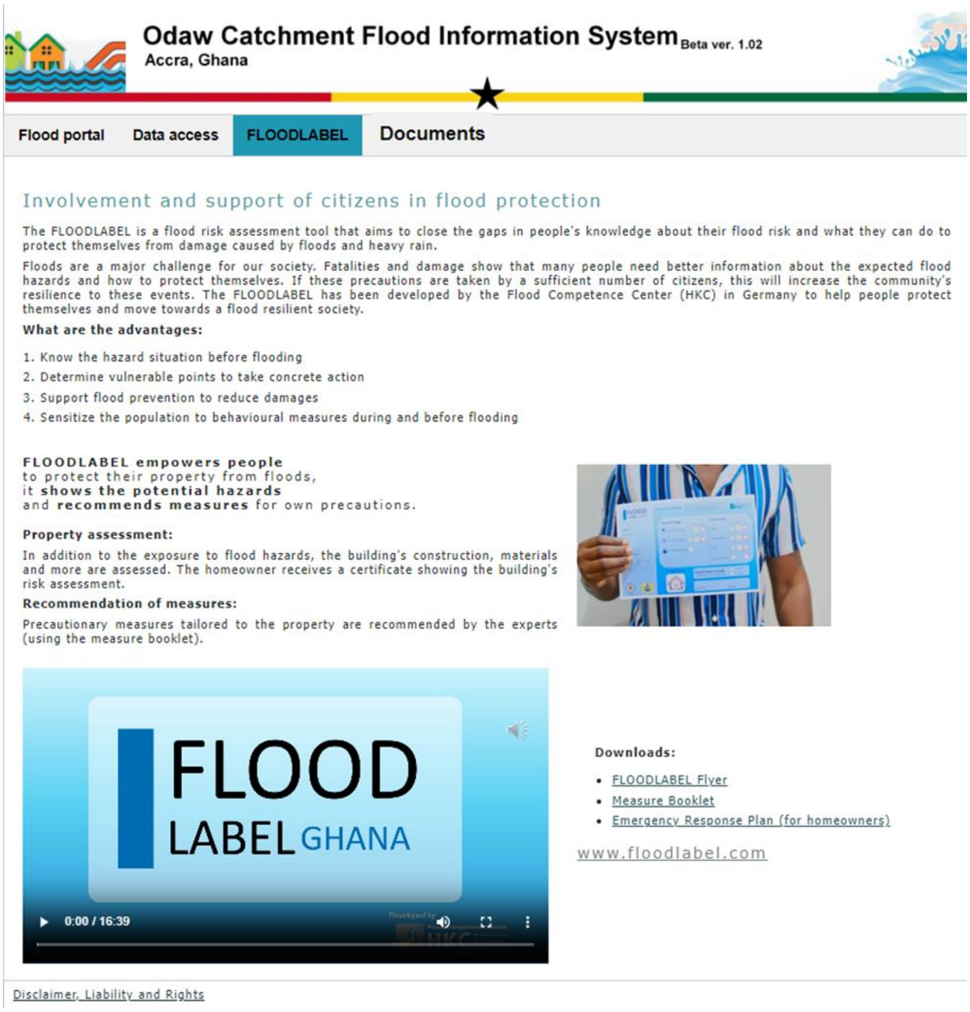


Figure 3: FLOODLABEL component

1.4 Documents

In this component (Fig. 3), the Document presents a comprehensive collection of documentation encompassing the varied products generated within the PARADeS project.

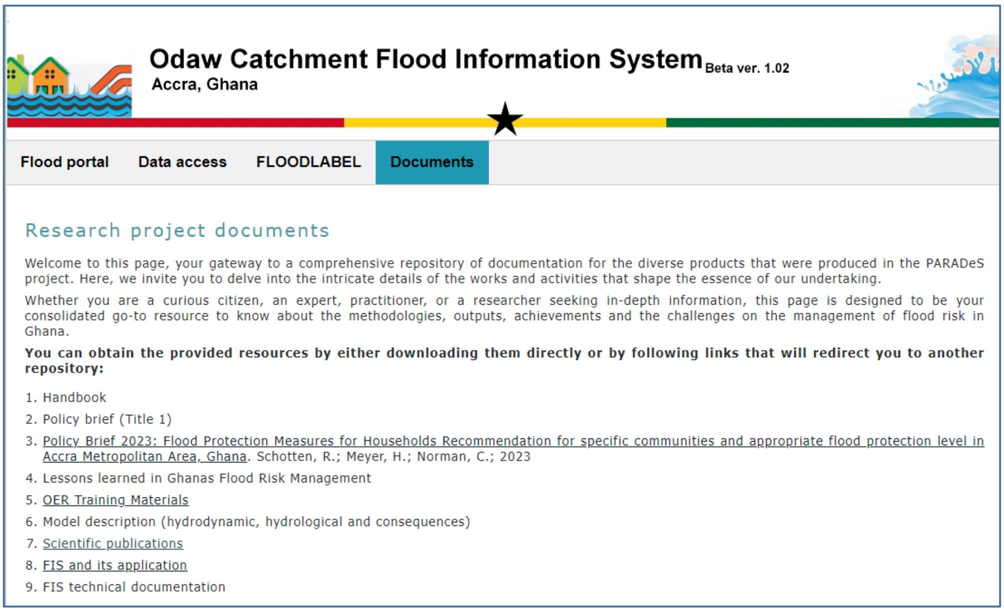


Figure 4: Document component

2. Conceptual and Technical design

In an era where climate change is leading to increasingly severe weather events, the need for efficient FIS has never been greater. A well-designed FIS web portal can serve as a crucial tool for providing accurate and timely flood-related data to the public, emergency responders, and decision-makers. The presented conceptual and technical design outlines the key mechanisms and functionalities of a FIS Web Portal to address this pressing need.

Conceptual design:

The conceptual design was further developed with stakeholder inputs in workshop setting to integrate their needs and requirements in the development of the FIS. The following criterial for the design was conceptualised based from the perception of the technical experts and stakeholders.

1. User-Friendly Interface:
   - A responsive and intuitive web interface accessible from desktop and mobile devices.
   - User-friendly navigation with an emphasis on accessibility for all users
   - A clean and modern design to enhance user experience and engagement.

2. Data Integration:
   - Integration of data from various sources (input and output)
   - Geographic Information System (GIS) integration for visualizing flood-prone areas and critical infrastructure.

3. Data Visualization:
   - Interactive maps with layers for displaying current and future flood conditions.
   - Location of important information (i.e critical infrastructure, safe havens)

4. Public Information and Education:

- Educational resources, including articles, videos, and infographics, to raise awareness about flood risks and safety measures.

5. Data Sharing and Collaboration:
   - Tools for local authorities, emergency services, and researchers to download and share data and products

6. Scalability:
   - Scalable architecture to accommodate new flood product data.
   - Interoperability using standards to publish data sets.

Technical design:

The technical design made use of open-source software's proven to be stable and secure. The idea is to make use of a spatial database system to store and share model results. Apart from the database system, the software system is composed of the model that produce the results of a simulations and a web-map-based user interface.

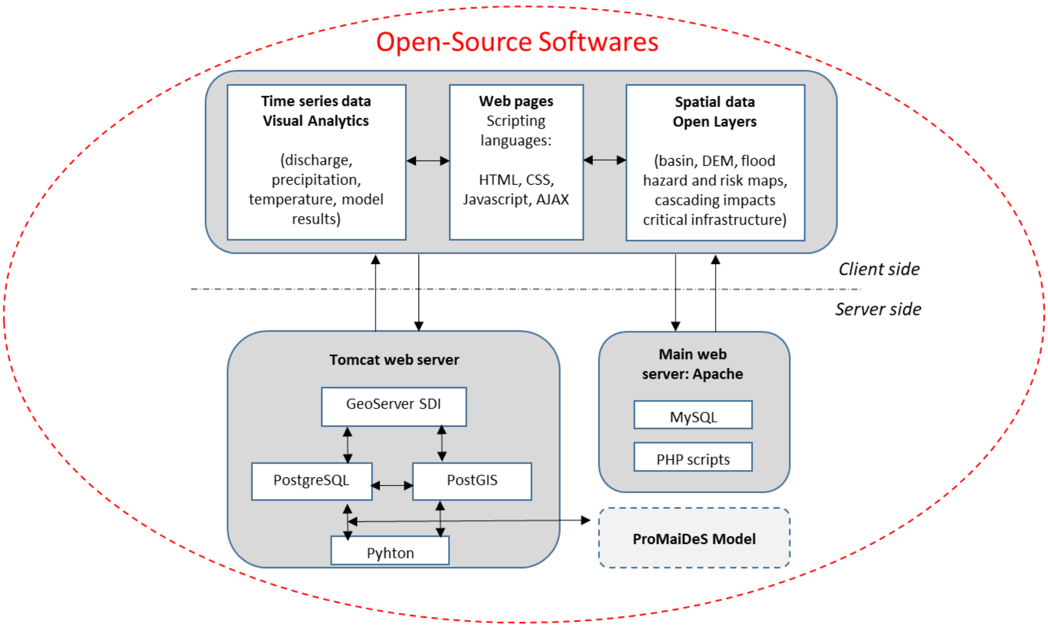


Figure 5: FIS technical design

3. Hardware, Software and Scripting languages

3.1 Hardware requirements

The first version of the FIS was wet-up with the following hardware specifications:

- Processor Intel(R) Core(TM) i7-9700K CPU @ 3.60GHz 3.60 GHz
- Installed RAM 32.0 GB (31.8 GB usable)
- System type 64-bit operating system, x64-based processor
- HDD 500 GB (used for FIS <50 GB)

3.2 Software- download and installation

The following are the software's used for the first version of the FIS with links to download and installation guide.

- Operating System:

  Windows 11 (We do not expect any problems for the FIS to run on other Windows version)

- HTTP Web server:

  Apache HTTP server- an open-source cross-platform web server software, is freely available and distributed under the Apache License 2.0. It is crafted and supported by a collaborative community of developers operating within the framework of the Apache Software Foundation.

  <u>Download and installation:</u>

  The first FIS version used the XAMPP package that contains the web Apache server, MySQL database, JAVA Tomcat.

  https://www.apachefriends.org

  The following XAMPP version was used:

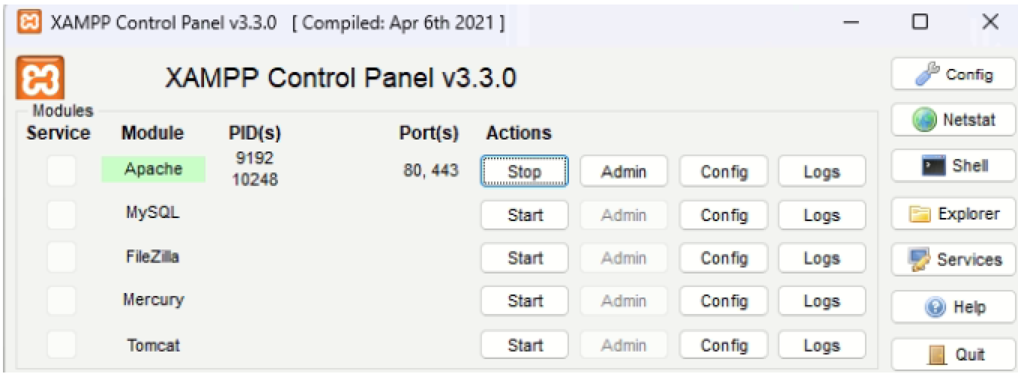


Figure 6: Web server Apache XAMPP version

- PostgreSQL:

  PostgeSQL were used to save the flood model results.

  PostgreSQL, commonly referred to as Postgres, stands as a free and open-source relational database management system with a focus on extensibility and adherence to SQL standards. Initially named POSTGRES, it evolved as a successor to the Ingres database, originating from the University of California.

  <u>Download and installation:</u>

  https://www.dbvis.com/database/postgresql

  Note: The ProMaIDes software package includes the PostgreSQL

- GeoServer:

  GeoServer were used to access and format the model results from the PostgreSQL to display in the FIS portal.

  GeoServer is an open source software server that allows users to share and edit geospatial data. It is written in Java and is designed to serve data using open standards such as Web Map Service (WMS), Web Feature Service (WFS), and Web Coverage Service (WCS). GeoServer can connect to various data sources, such as spatial databases, shapefiles, and remote WMS servers, and publish them as web services that can be accessed by clients such as web mapping applications. It provides a web-based interface that allows users to configure and manage the server, including defining data sources, creating and configuring layers, setting up security, and generating maps. It supports a wide range of vector and raster data formats and allows for advanced styling and labeling of map features.

  GeoServer is widely used in various domains such as environmental monitoring, urban planning, disaster management, and agriculture. It is considered to be a flexible and scalable solution for sharing and serving geospatial data over the web.

  Download and installation:

  https://geoserver.org

  The following GeoServer version was used:

Figure 7: FIS GeoServer version

- Openlayers:

  Designed to display map data on web browsers OpenLayers is a JavaScript library, available as open-source. This library offers an application programming interface (API) that facilitates the creation of feature-rich, web-based geographic applications, akin to the functionality provided by Google Maps and Bing Maps.

  Information:

https://openlayers.org

The following Openlayer version was used:

<script src="https://cdn.jsdelivr.net/npm/ol@v7.2.2/dist/ol.js"></script>

3.3 Scripting languages

Scripting languages that were used for the FIS are:

a) HyperText Markup Language (HTML)

HTML is the standard markup language used to create and design documents on the World Wide Web. It provides a structure for web content, allowing developers to define and organize elements on a web page. HTML consists of a set of tags and attributes that define the structure and presentation of content, such as text, images, links, forms, and more.

b) CSS (Cascading Style Sheets),
CSS is a style sheet language used in web development to control the presentation and layout of HTML documents. CSS allows developers to define the visual appearance of web pages by specifying the styling of elements such as text, fonts, colors, spacing, and positioning.

c) Asynchronous JavaScript and XML (AJAX)

AJAX is a set of web development techniques used to create asynchronous web applications. In simpler terms, it allows parts of a web page to be updated asynchronously by exchanging small amounts of data with the server behind the scenes. This enables a more dynamic and responsive user experience.

4. Example web-map application and access to model results

In this example application you may choose to install the XAMPP with the Apache server depending if you want it to be accessible in the internet or not. The Geoserver and Postgresql can be installed for further self-exploration.

For demonstration, the spatial data published are hosted in a remote Geoserver. It is advisable to set-up your Geoserver if you want to publish the ProMaIDes model results project.

Section 4.1 will first provide you an example on setting-up a simple web-map application and Section 4.2 guides you on how to connect with the PostgreSQL database using the Geoserver to gain access and publish your ProMaIDes model results in the web-map application.

4.1 Web-map application

The core of the web-map application is the Openlayers API. Here we demonstrate the setting up of a simple map portal. You can explore Openlayers for some other functionalities in their website https://openlayers.org.

The following steps and scripts are a snippet of the Ghana-FIS flood portal.

a) Step 1 (optional):

- If you want to publish example portal into the world wide web run the Apache server in the XAMPP control panel

Figure 8: XAMPP Control panel- Apache Server

- Create a folder “FIS_example” in the htdocs installation path of XAMPP. i.e. C:\xampp\htdocs\

b) Step 2:

- Download the zip file “**Example_codes**”
- Copy the folder “**Web_Map**” to the “**FIS_example**” folder
- The “Web_Map” folder contains the files “index.hmtl” and “main.js”

  Using the Notepad software (https://notepad-plus-plus.org/downloads/) you can explore and also edit the script

  The **index.html** contains the script html and css to create the visual layout and content of the homepage., here it mainly displays the map layout. In the script, a description is provided as a comment with the following tag “<!--- … --->”.

  The **main.js** contains the javascripts to set-up and customised the map. In the script, a description is provided as a comment with the following tag “/** … **/”. Please refer to https://openlayers.org for the detailed description of the scripts written.

c) Step 3:

- Display the web-map in the web-browser using the following option

  - Option 1: Double click on the index.html
  - Option 2: Run your Apache server through XAMPP and display the portal using the following URL in your web browser

    http://localhost/FIS_example/Web_Map

OR using your IP address

i.e. http://131.220.xxx.xxx/FIS_example/Web_Map

Figure 9: Example Web-map

d) Step 4 (example editing):

- Now try to enlarge the map by adjusting the styles of the map height and width in the “index.html”.
- Reload the portal and see if there are any changes

4.2 Access to model results

This section guides you on connecting your Geoserver to the PostgreSQL to access and publish your ProMaIDes model results in the web-map application. For demonstration we will provide some screenshots of the example set-up. In your project, you have to use the specific IP address of your Geoserver and credentials to access your local PostgreSQL.

a.) Connecting the Geoserver to the PostgreSQL ProMaIDes model results

You may skip this part if you only want to explore the publishing of the model results in the example scripts (html and javascript)

- Step 1: Log-in to the Geoserver as Admin- http://localhost:8080/geoserver

- Step 2: Create a new "Workspace". In this example we named it Ghana_Accra

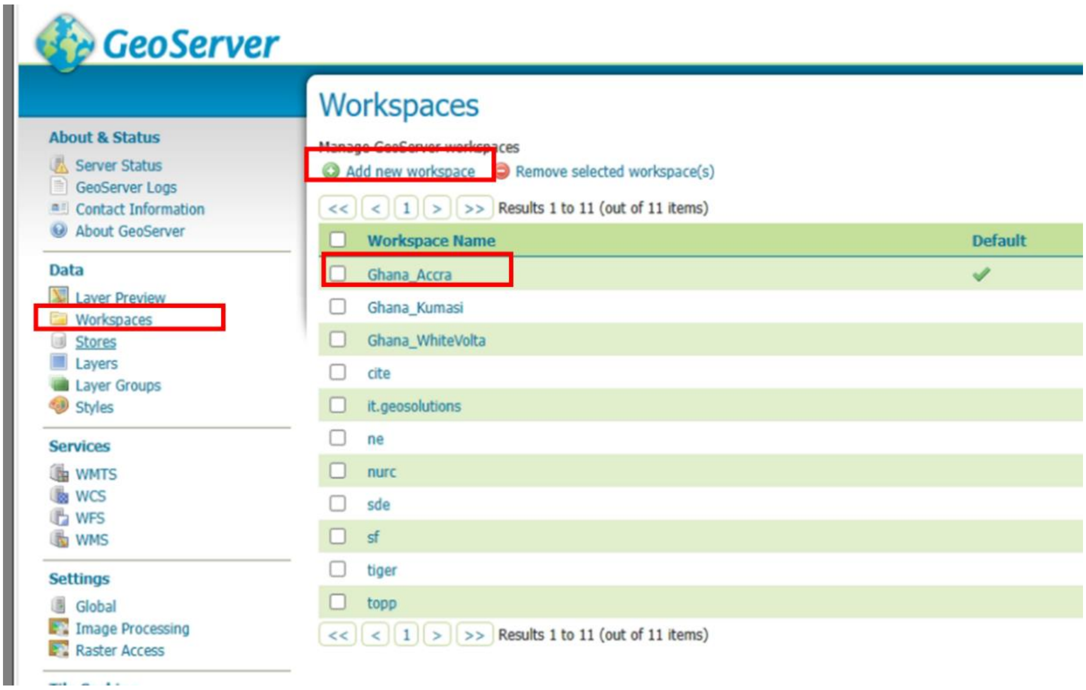


Figure 10: GeoServer- Workspace

- Step 3: Create a new "Store". This connects your GeoServer to the PostgreSQL model results.

  Presented in Figure 12 are the information needed to set-up the connection. You can use your local PostgeSQL IP address and its credentials

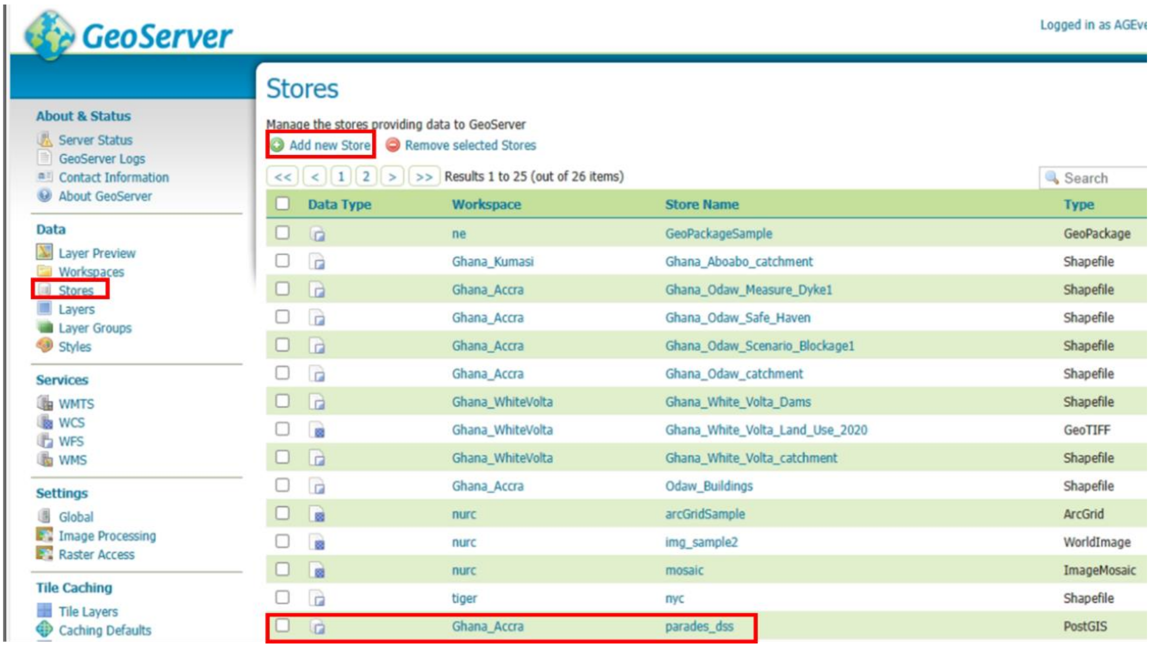


Figure 11: GeoServer Stores

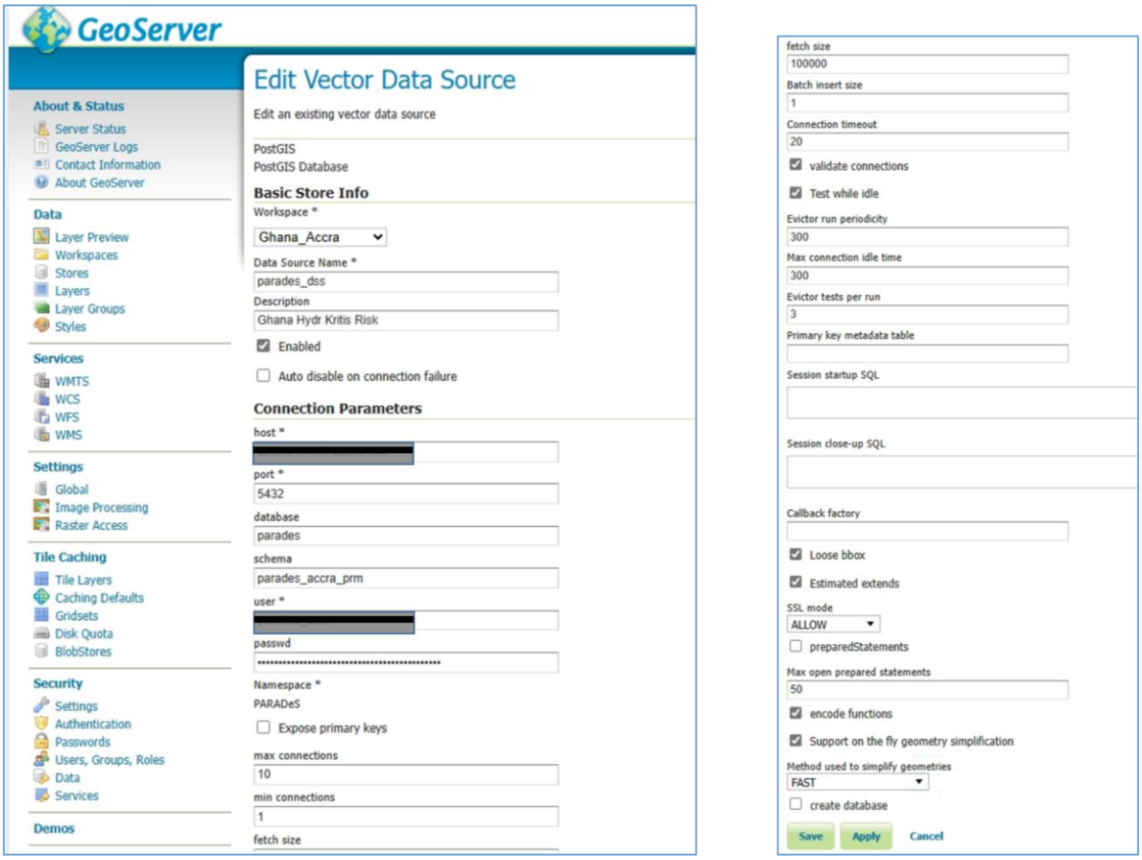


Figure 12: GeoServer Stores Information

- Step 4: Create a new "Layer". This connects you to the specific layer model results

Figure 13: GeoServer Layers

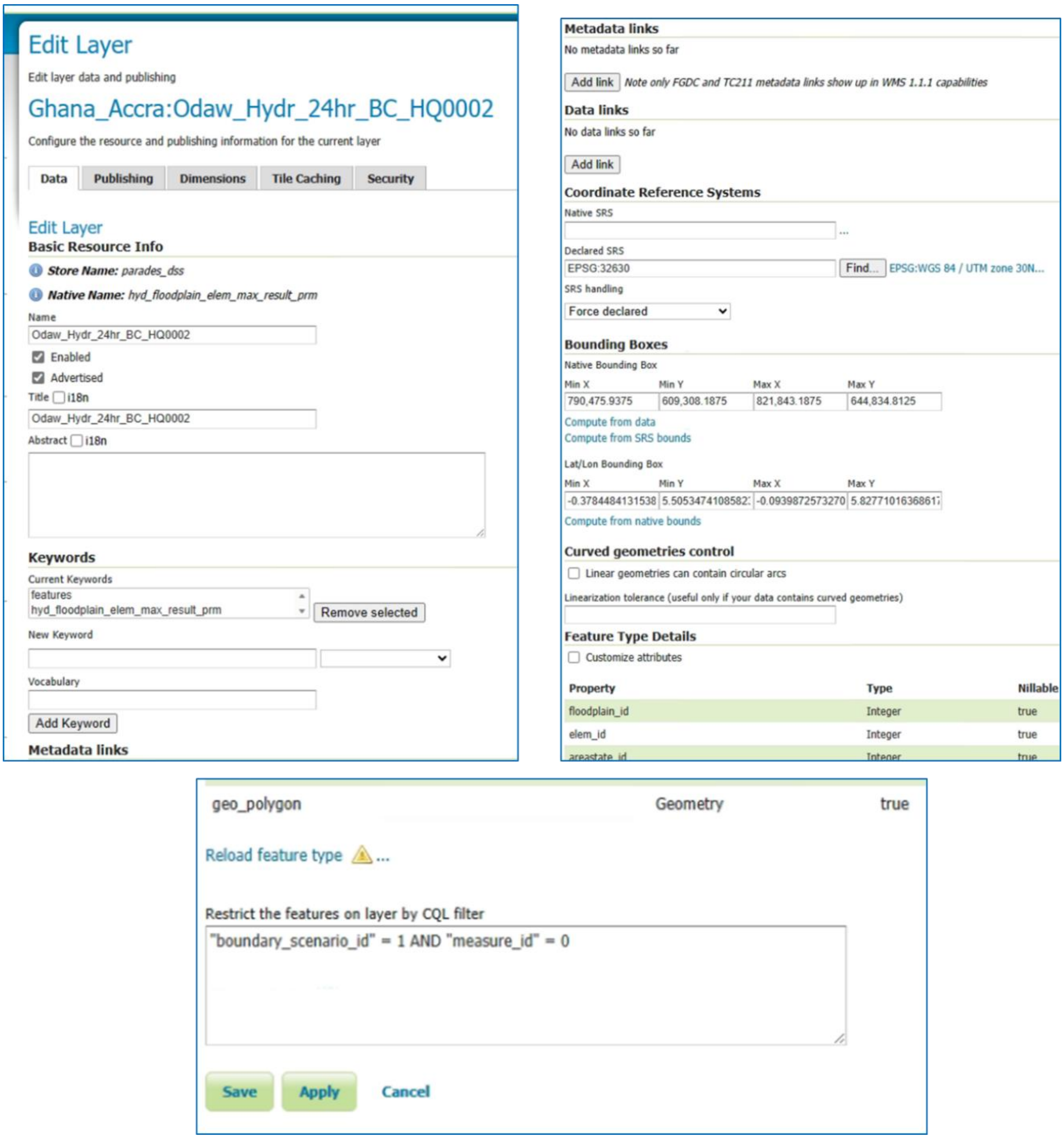


Figure 14: GeoServer Layer Data Information. *Note: The bottom figure is the identifier what layer to fetch from the PostgreSQL. This is the identifier when setting up the scenario and measure id in the ProMaIDes model*

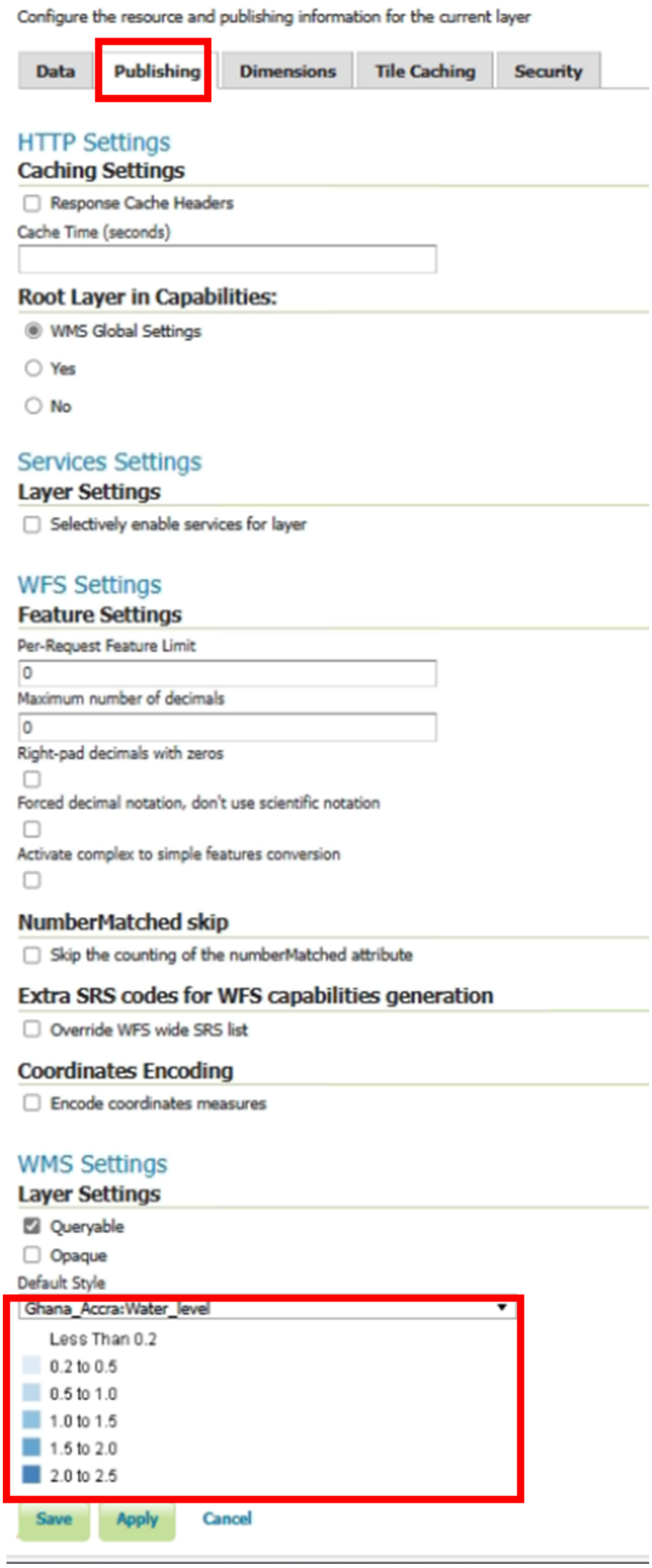


Figure 15: GeoServer Layer Publishing Information. *Note: You may define your own styles in the "Styles" menu of Geoserver. Refer to this documentation on styling- https://docs.geoserver.org/main/en/user/styling/sld/introduction.html*

- Step 5: View the published data in the "Layer Preview"

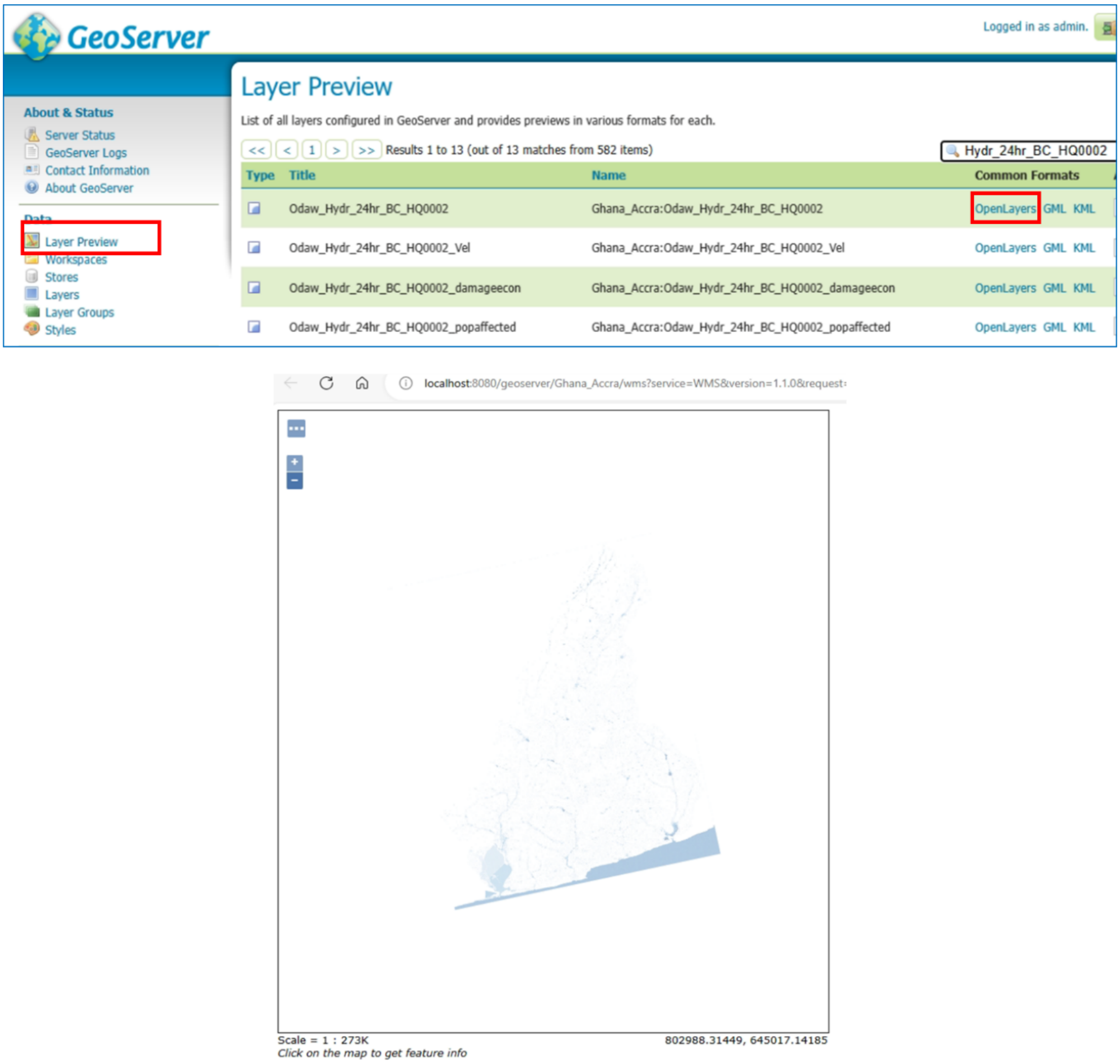


Figure 16: GeoServer Layer Preview

b.) Publishing the model results in the web-map application

- Step 1:
  - ➢ In the downloaded zip file "[**Example_codes**]" copy the folder "**Web_Map_Model**" to the "**FIS_example**" folder
  - ➢ The "Web_Map_Model" folder contains the files "index.hmtl", "main.js" and "Var_Hazard.js"

    Using the Notepad software (https://notepad-plus-plus.org/downloads/) you can explore and also edit the script

    The **index.html** contains the same html and css script presented in Section 4.1. Additions are mainly in the javasctipts file.

The **main.js** contains the javascripts to set-up and customised the map. In the script, a description is provided as a comment with the following tag "/** … **/". Please refer to https://openlayers.org for the detailed description of the scripts written.

The **Var_Hazard.js** contains the information of the scenario stored in a variable array. This comprises the layer name of the data in the Geoserver, the rainfall depth, water level and discharge model results.

- Step 2: Display the web-map in the web-browser using the following option

  - Option 1: Double click on the index.html
  - Option 2: Run your Apache server through XAMPP and display the portal using the following URL in your web browser

    http://localhost/FIS_example/Web_Map_Model

    OR using your IP address to make it accessible over the web

    i.e. http://131.220.xxx.xxx/FIS_example/Web_Map_Model

Figure 17: Example Web-map publication pf model results

5. Maintenance recommendation

To ensure the optimal functioning of the FIS portal, it is advisable to stay informed about potential updates to the OpenLayers libraries and GeoServer for maintenance purposes. Implementing such upgrades could lead to the obsolescence of the current version employed in the FIS. If updates to the OpenLayers libraries are necessary, it is recommended to verify their compatibility with both the GeoServer version and, consequently, the PostgreSQL database.

For the upgrade of the GeoServer please refer to this documentation-
https://docs.geoserver.org/2.23.x/en/user/installation/upgrade.html

Furthermore, it is crucial to create a backup of the FIS files and database to mitigate any unforeseen issues that may arise during the maintenance process.

Project Partners

Sponsored by the Federal Ministry of Education and Research

Citation reference:

Prof. Dr. Mariele Evers
University of Bonn

Dr. Sylvia Kruse
University of Freiburg

Prof. Dr.-Ing. Daniel Bachmann
University of Applied Sciences in Magdeburg

Dipl.-Hydrology Georg Johann
Flood Competence Center