An INTERNSHIP PROJECT REPORT on INTEGRATED WATER MANAGEMENT SYSTEM

Submitted in partial fulfilment of the requirements of the degree

Bachelor of Technology

In

COMPUTER SCIENCE AND ENGINEERING

By

CHALLA SIVAKUMARI (R170615)

Under the supervision

of B.Lingamurthy

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING,

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE

TECHNOLOGIES, RK VALLEY CAMPUS,

MAY 2023.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING, RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES, RK VALLEY CAMPUS, MAY2023.

CERTIFICATE

This is to certify that the project report entitled **INTEGRATED WATER MANAGEMENT SYSTEM (INTERNSHIP REPORT)** submitted by **CHALLA SIVAKUMARI (R170615)** in partial fulfillment of the requirement for the award of Bachelor of Technology in Computer Science and Engineering is a record of bonafide project work carried out under my supervision during the academic year 2022-23.

The report hasn't been submitted previously in part or in full to this or any other university or institution for the award of any degree.

B.Lingamurthy
Software Engineer,
Department of CSE,
RGUKT, RK VALLEY

Mr. N. Satyanandaram
Head of Department,
Department of CSE,
RGUKT, RK VALLEY.

DECLARATION

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Signature

____

CHALLA SIVAKUMARI
ID: R170615

Date: _______________

ACKNOWLEDGEMENT

I am highly indebted to **Software Engineer Mr.B.Lingamurthy** for his guidance and constant supervision as well as for providing necessary information regarding the project and also for their kind cooperation, encouragement and their support in completing the project.

I would like to express my special gratitude and thanks to our COMPUTER SCIENCE AND ENGINEERING branch **H.O.D Mr. N. Satyanandaram**
and **Director of RK Valley-RGUKT Prof. K.Sandhya Rani** for giving me such attention and time.

I have taken efforts in this project. However, it could not have been possible without the kind support and help of many individuals and RGUKT. We would like to extend my sincere thanks to all of them.

My thanks and appreciation also go to my colleagues in developing the project and people who have willingly helped me out with their abilities.

With Sincere Regards,
CHALLA SIVAKUMARI

Date: _______________

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

This project aims to provide valuable insights into various types of environmental data, including rainfall, groundwater level, river water level, inflow forecast, soil moisture, flood forecasting, and reservoir water level. The project collects data from various sensors and websites and analyzes it using various machine learning algorithms to provide insights into the current and future state of the environment.

The project's main objective is to provide real-time and accurate data on the various environmental parameters to help decision-makers make informed decisions. The project collects data from various sources, including government websites, rainfall stations, weather stations, and ground-based sensors. The data collected is stored in a database and analyzed using various machine learning algorithms to provide insights into the current state of the environment.

The project also uses predictive modeling techniques to forecast future environmental conditions, such as flood levels and reservoir water levels. The predictions are made based on historical data, weather forecasts, and other environmental factors.

In conclusion, this project is an essential tool for environmental monitoring and management, providing valuable insights into the state of the environment and helping decision-makers make informed decisions.

CHAPTER 1
INTRODUCTION

Water management is a crucial aspect of sustainable development, and with increasing pressure on water resources due to climate change, population growth, and economic development, the need for effective water management systems is becoming increasingly important. This project aims to provide real-time insights into various water-related data, including rainfall, groundwater level, river water level, inflow forecast, soil moisture, flood forecasting, and reservoir water level, by collecting data from sensors and websites. The data is then processed and analyzed to generate dashboards, heat maps, graphs, and tables that provide valuable insights into water resources management. The project also includes several Decision Support Systems (DSS) products such as Flood forecasting, Reservoir Management & planning, Canal Management System, and Water audit and budgeting.

1.1 Motivation

The motivation behind this project is to address the growing need for effective water management systems due to climate change, population growth, and economic development. Water is a scarce resource, and with the increasing demand for water, there is a need for effective and efficient water management systems.

1.2 Features

- **Real-time data collection:** The system collects real-time data from sensors and websites related to water resources such as rainfall, groundwater level, river water level, inflow forecast, soil moisture, flood forecasting, and reservoir water level.
- **Data processing and analysis:** The collected data is processed and analyzed to generate dashboards, heat maps, graphs, and tables, which provide valuable - insights into water resource management.

- **Decision Support Systems (DSS):** The system includes several DSS products such as Flood forecasting, Reservoir Management & planning, Canal Management System, and Water audit and budgeting, which can assist in efficient water resource management.
- **Interpolation and aggregation**: The system uses interpolation and aggregation techniques to provide data at various levels, including village, mandal, district, and state levels.
- **Scalability:** The system is designed to be scalable and can handle large amounts of data with ease.
- **User-friendly interface:** The system provides a user-friendly interface that allows decision-makers to access real-time data and DSS products easily .
- **Integration with third-party systems:** The system can be easily integrated with third-party systems to provide a comprehensive water resource management solution.
- **Robust architecture:** The system is designed using Springboot using Java, Kafka messaging queue, Quartz scheduler, Flink, Cassandra, Postgres, Docker, Maven, Git, Node JS, Angular, and Geoserver, which makes the system robust and reliable.
- **High-speed data processing:** The system uses Flink, a high-speed stream processing framework, to process data in real-time.
- **Security:** The system is designed with security in mind, and all data is stored securely and encrypted to ensure data integrity and confidentiality.

Overall, the key features of this project make it a comprehensive water resource management system that can assist decision-makers in making informed decisions regarding water resource management.

CHAPTER 2
SYSTEM ANALYSIS

Analysis is the detailed study of the various operations performed by a system and their relationships within and outside of the system. A key question is: What must be done to solve the problem? One aspect of analysis is defining the boundaries of the system and determining whether or not the candidate system should consider other related systems During analysis, data are collected on the available files, decision points, and problems handled by the present system.

2.1 Existing System

The existing system lacks a comprehensive approach to water resource management, with limited real-time data available to decision-makers. The system also lacks the necessary tools for data analysis and forecasting, which can result in poor decision-making.

The existing system lacks a comprehensive approach to water resource management, with limited real-time data available to decision-makers. The system also lacks the necessary tools for data analysis and forecasting, which can result in poor decision-making.

2.2 Proposed System

The proposed system is a comprehensive platform for providing real-time insights into various water parameters. The system takes data from sensors and websites and aggregates and analyzes it to generate dashboards, heat maps, graphs, tables, and other visualizations. The platform is designed to handle large volumes of data and provide timely and accurate insights to decision-makers.

The system uses advanced data analytics techniques, such as interpolation and aggregation, to provide accurate and reliable insights. The system is built using modern technologies such as Springboot using java, Kafka messaging queue, Quartz scheduler, Flink, Cassandra, Postgres,Docker, Maven, Git, Node JS, Angular, Geoserver, etc., to ensure that it is scalable, reliable, and can handle large volumes of data.

CHAPTER 3
REQUIREMENT ANALYSIS

The project involved analyzing the design of a few applications so as to make the application more users friendly. To do so, it was really important to keep the navigations from one screen to the other well-ordered and at the same time reducing the amount of typing the user needs to do. This also includes maintaining the flow of the application.

3.1 Requirement Specification

3.1.1 Functional Requirements

- Data collection from sensors and websites
- Real-time data processing
- Dashboards, heat maps, graphs, and tables generation
- Flood forecasting
- Reservoir management & planning
- Canal management system
- Water audit and budgeting

3.1.2 Hardware Requirements

- Sensors for data collection
- High-speed internet connectivity
- Computer servers for data processing and storage
- Processor: A multi-core processor with a clock speed of at least 2.5 GHz.
- RAM: At least 16 GB of RAM is recommended for running the system in production. However, the actual RAM requirements will depend on the amount of data being processed and the number of concurrent users accessing the system.
- Storage: The system requires a minimum of 500 GB of storage for storing data. However, the actual storage requirements will depend on the amount of data being processed and the duration for which the data needs to be stored.
- Network: The system requires a reliable high-speed internet connection to ensure smooth data transfer.

- Operating System: The system can run on any operating system that supports Docker containers. However, Linux-based operating systems are recommended for running the system in production.
- Server: A dedicated server is recommended for running the system in production. The server should have sufficient resources to handle the load generated by the system.
- Backup and Recovery: The system should be backed up regularly to ensure data integrity and availability. A backup solution should be implemented to ensure that the system can be quickly restored in case of data loss or system failure.

3.1.3 Software Requirements

- Operating System : Windows 7 and Above, Linux/Ubuntu 14 and above
- Packet Management Tool : Node Package Manager (NPM)
- FrontEnd Technologies : HTML, CSS, Bootstrap, JavaScript, AngularJS
- Backend Technologies : NodeJS, ExpressJS, Java
- Database : PostgreSQL, Cassandra, Geoserver
- Frameworks : SpringBoot
- Dependency management tools : Maven
- Source code management : Git
- Container management : Docker
- Cloud : AWS
- Streaming platform : Kafka messaging queue
- Job scheduler : Quartz Scheduler
- Data Processing : Flink

3.2 Technologies Used

3.2.1 HTML[1]

It is a markup language for formatting and displaying web documents and web pages. It gives basic structure to the webpage without any styling. HTML elements tell the browser how to display the content. It can be assisted by technologies such as Cascading Style Sheets and scripting languages such as JavaScript for styling and functionality

3.2.2 CSS[2]

It gives styling for the web pages created by HTML. It gives ‘ look and feel ’ to the website.

3.2.2.1 Types of CSS

- Inline CSS (Using styles as attributes in html elements)
- Internal CSS (Including a separate Style tag in html document)
- External CSS (Using external file for styling)

3.2.3 Bootstrap[3]

Bootstrap is a CSS framework which helps in developing web pages very faster and with little efforts. Helps to customize the CSS properties. Used for developing responsive and mobile- first websites. Components like navbar, carousel, utility, cards, dropdowns, buttons etc.

3.2.4 JavaScript

JavaScript is used to develop interactive web applications. Used to develop Dynamic websites. JavaScript is the programming language of the Web. Responsible for performing actions in a website.

3.2.5 AngularJS[4]

AngularJS is a discontinued free and open-source JavaScript-based web framework for developing single-page applications. It was maintained mainly by Google and a community of individuals and corporations

3.2.6 NodeJS[5]

NodeJs is an open source server Environment that allows us to run JavaScript on the Server(outside a web browser). It is an Event driven architecture capable of Asynchronous Input/Output. It is a JavaScript runtime built on Chrome's v8 Javascript Engine. Runs equally well on All platforms(Windows,mac and linux)

3.2.7 ExpressJS[6]

Express Js is the most popular , free and Open-source server-side web application framework for NodeJs. It provides a strong set of features to build web applications quickly and easily. It provides an Integrated environment to facilitate rapid development of Node based web applications. Express Js allows to setup middlewares to respond to HTTP requests. Express provides methods to specify what function is called for a particular HTTP verb(GET , PUT , POST, DELETE) and URL pattern('route').

3.2.8 PostgreSQL[7]

PostgreSQL is a powerful open-source relational database management system (RDBMS) that uses and extends the SQL language. It is widely used as a backend database for web applications, mobile applications, and analytics platforms due to its reliability, stability, and high-performance features. PostgreSQL supports a variety of data types, including numeric, string, date and time, boolean, and spatial data. It also offers advanced features such as transaction management, multi-version concurrency control, table partitioning, and full-text search. PostgreSQL is free and can be used on a variety of operating systems, including Linux, Windows, and macOS.

3.2.9 Cassandra

Cassandra is an open-source distributed database management system that is designed to handle large amounts of data across many commodity servers, providing high availability with no single point of failure. It was developed by Facebook initially and is now maintained by the Apache Software Foundation.

Cassandra is a NoSQL database, which means that it does not use the traditional relational model of databases that organize data into tables with strict schema definitions. Instead, Cassandra uses a column-family data model, which allows for flexible and dynamic schema design. Data is stored in key-value pairs, where the key is used to uniquely identify each row of data, and the value is a collection of columns that represent the data attributes.

3.2.10 Kafka

Kafka is an open-source distributed streaming platform that was originally developed by LinkedIn and is now maintained by the Apache Software Foundation. Kafka is designed to handle real-time data feeds and data processing, providing a fast, scalable, and reliable solution for processing and analyzing data streams.

3.2.11 Flink

Apache Flink is an open-source distributed stream processing framework that was designed to handle real-time data processing applications. Flink supports both batch and streaming data processing, and provides a powerful, flexible, and fault-tolerant system for processing data streams.

3.2.12 Quartz-Scheduler

Quartz is an open-source job scheduling framework that can be integrated into Java applications. It allows developers to schedule jobs or tasks to run at specific times or intervals, and provides a range of features for managing and monitoring those jobs.

3.2.13 Geo Server

GeoServer is an open-source server for sharing and publishing geospatial data. It provides a web interface for managing and publishing geospatial data in various formats, including Open Geospatial Consortium (OGC) standards such as Web Map Service (WMS), Web Feature Service (WFS), and Web Coverage Service (WCS).

3.2.14 Docker

Docker is an open-source platform for building, deploying, and running applications in containers. Containers are lightweight, portable, and self-contained environments that allow developers to package their applications and dependencies into a single unit that can be easily deployed across different environments.

3.2.15 AWS-EC2 Instance

Amazon Elastic Compute Cloud (EC2) is a web service that provides resizable compute capacity in the cloud. EC2 allows customers to rent virtual servers, known as instances, on which they can run their applications. EC2 instances can be launched on-demand, which means they can be started and stopped as needed, and customers only pay for the computing resources they use.

3.2.16 Spring Boot

Spring Boot is an open-source Java framework that provides a rapid application development environment for building web applications. It is built on top of the popular Spring Framework and provides a set of tools and libraries that help developers get up and running quickly with minimal configuration.

3.2.17 Maven

Apache Maven is a build automation tool used primarily for Java projects. It is an open-source software that helps to manage project dependencies, build processes, and project documentation.

Maven uses a declarative approach to build processes, where a project's dependencies, configuration, and other aspects of the build process are described in a configuration file called a pom.xml. This file defines the project's dependencies, the build process, and the output artifacts.

CHAPTER 4
SOFTWARE ARCHITECTURE

4.1 Event Driven Architecture

The event-driven architecture pattern is a popular distributed asynchronous architecture pattern used to produce highly scalable applications. It is also highly adaptable and can be used for small applications as well as large, complex ones. The event-driven architecture is made up of highly decoupled, single-purpose event processing components that asynchronously receive and process events.

Event channels are conduits in which events are transmitted from event emitters to event consumers.

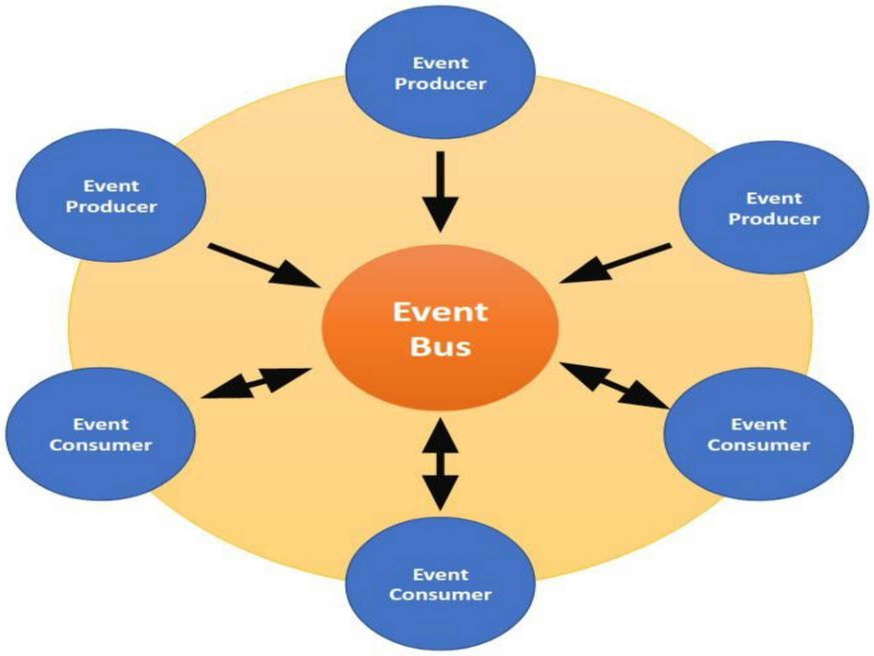


Figure 4.1: Generic Design of an event-driven architecture

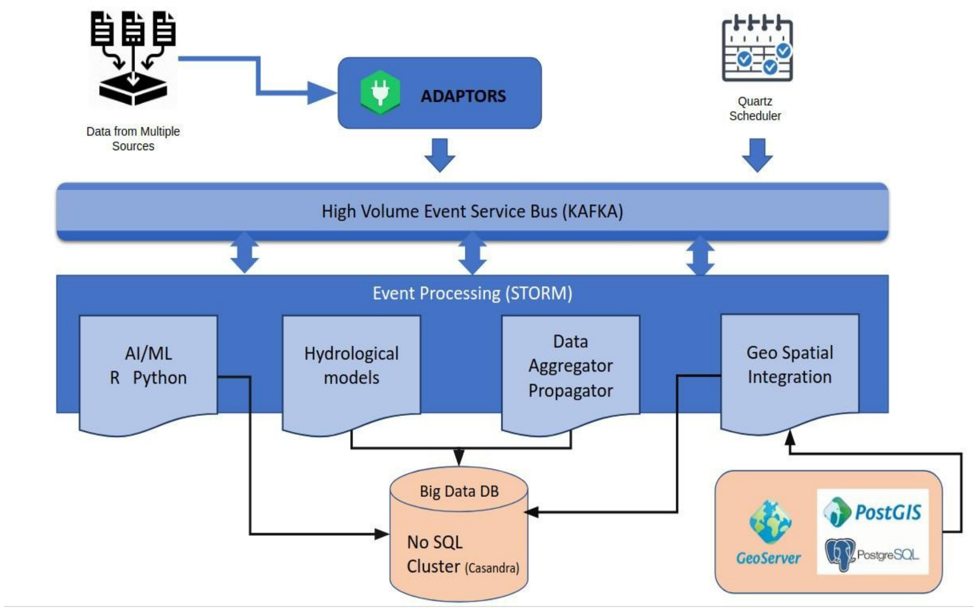


Figure 4.1.2: Event driven backend architecture

4.2 Web Application Overview/Architecture

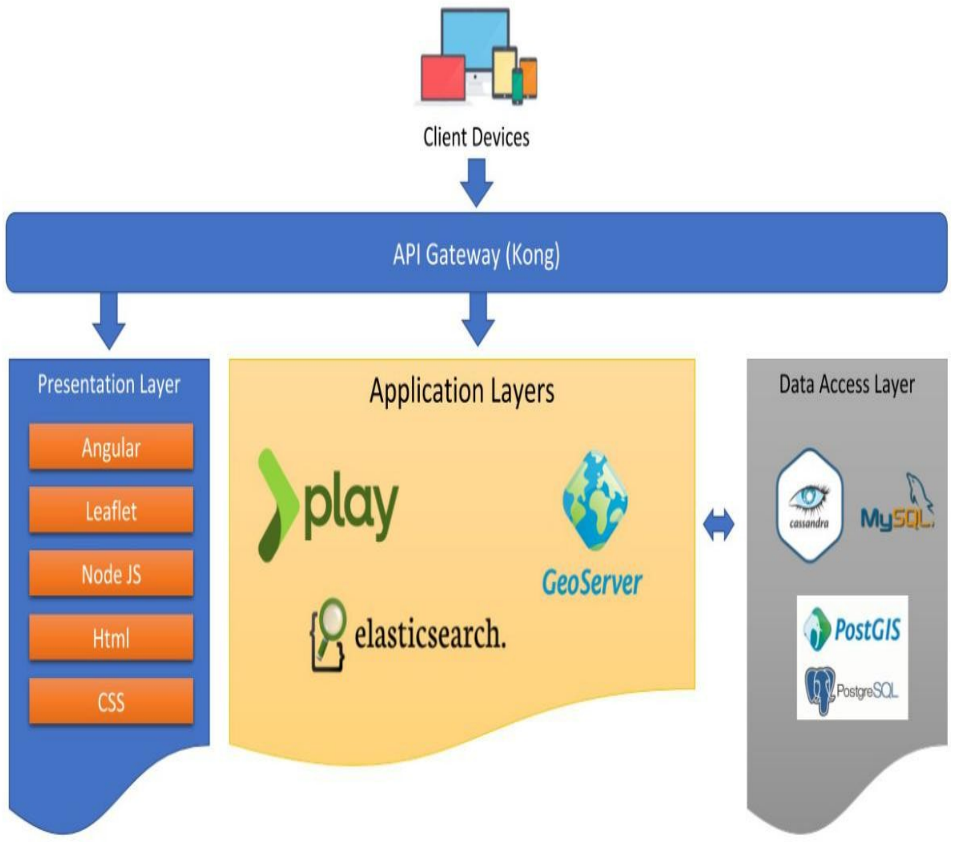


Figure 4.2 Web Application Architecture

CHAPTER 5
SOFTWARE ENVIRONMENT

5.1 Intellij IDEA(IDE)

IntelliJ IDEA is a popular Integrated Development Environment (IDE) developed by JetBrains. It is primarily used for programming in Java, but also supports several other programming languages like Kotlin, Scala, Groovy, and Python.

IntelliJ IDEA provides advanced features like code completion, syntax highlighting, code refactoring, debugging, and unit testing. It offers advanced code analysis, navigation features, code completion, refactoring tools, debugger and unit testing tools, integration with version control systems, support for multiple build systems, and integration with various web frameworks and application servers. It is available in both community and ultimate editions, with the ultimate edition offering additional features. IntelliJ IDEA is known for its user-friendly interface and high productivity features for developers.

5.2 Embedded Tomcat Server

Embedded Tomcat server refers to the integration of Apache Tomcat, a popular web server and servlet container, within a Java application. This allows the application to run as a self-contained, standalone executable, without requiring a separate Tomcat installation or deployment to an external server.

An embedded Tomcat server is often used in applications that require a lightweight, scalable, and easily deployable web server. The server can be configured programmatically within the application code, allowing developers to fine-tune the server settings to meet their specific requirements.

To use an embedded Tomcat server in a Java application, developers typically include the Tomcat libraries as dependencies in their project and configure the server programmatically within the application code. The server can be started and stopped within the application, and web applications can be deployed to the server dynamically at runtime.

Some benefits of using an embedded Tomcat server include faster development cycles, simplified deployment and distribution, and reduced operational costs. However, it may not be suitable for large-scale applications with high traffic volumes or complex requirements, which may require a dedicated, externally-managed Tomcat server.

5.3 YAML Files

YAML (YAML Ain't Markup Language) is a human-readable data serialization language that is often used for configuration files, data exchange, and data storage. It was designed to be easily readable by humans and is often used in modern software development practices.

YAML files use indentation and whitespace to represent data hierarchies, making it easier to read and write than other data serialization formats, such as JSON or XML. YAML files are often used for configuration files in software development, such as defining application settings or specifying build and deployment configurations.

YAML supports a wide range of data types, including strings, numbers, booleans, and null values, as well as more complex data structures such as maps, lists, and nested structures. It also supports comments, allowing developers to provide additional context or documentation within the configuration files.

YAML files are commonly used in software development frameworks and tools, such as Kubernetes, Ansible, and Docker, for defining configurations and orchestrating deployment and management tasks.

CHAPTER 6
IMPLEMENTATION

6.1 River Gauge Component Functionalities

6.1.1 River Gauge Definition

A river gauge is a device used to measure the height or depth of water in a river or other body of water. The data collected by river gauges is used to monitor water levels, track changes over time, and help predict floods or other natural disasters. Accurate river gauge data can help emergency responders and officials make informed decisions about evacuations, road closures, and other measures to protect people and property.

6.1.2 Scraping

Web scraping, is the process of extracting information from websites using automated tools. It involves writing code to visit a website, inspect its HTML structure, and extract the relevant data. The extracted data can be used for various purposes, such as research, analysis, or building applications. Web scraping can be done manually, but it is typically done using automated tools such as web crawlers, which systematically browse the web and collect data from multiple websites.

When scraping data, it is important to adhere to ethical and legal guidelines, such as respecting the website's terms of use, avoiding excessive or disruptive requests, and obtaining explicit consent if necessary. Additionally, it is important to ensure the quality and accuracy of the scraped data, as well as protect the privacy and security of any personal information that may be involved.

6.1.3 Aggregation

Aggregation is the process of combining multiple data points into a single summary value, either over time or over space, to gain a better understanding of the data. The types of data aggregations we are using in this project are spatial and temporal data aggregation.

6.1.3.1 Spatial Aggregation

Spatial aggregation refers to the process of collecting and combining data from lower-level geographic units into higher-level units. For example, in a village, data may be collected from
various rivergauge stations, sensors. This data can then be aggregated at the mandal level, which is a higher-level administrative division. The Mandal data can then be aggregated at the District level,Which is a higher-level administrative division from Mandal. The District data can then be aggregated at the State level,Which is a higher-level administrative division from District.

6.1.3.2 Temporal Aggregation

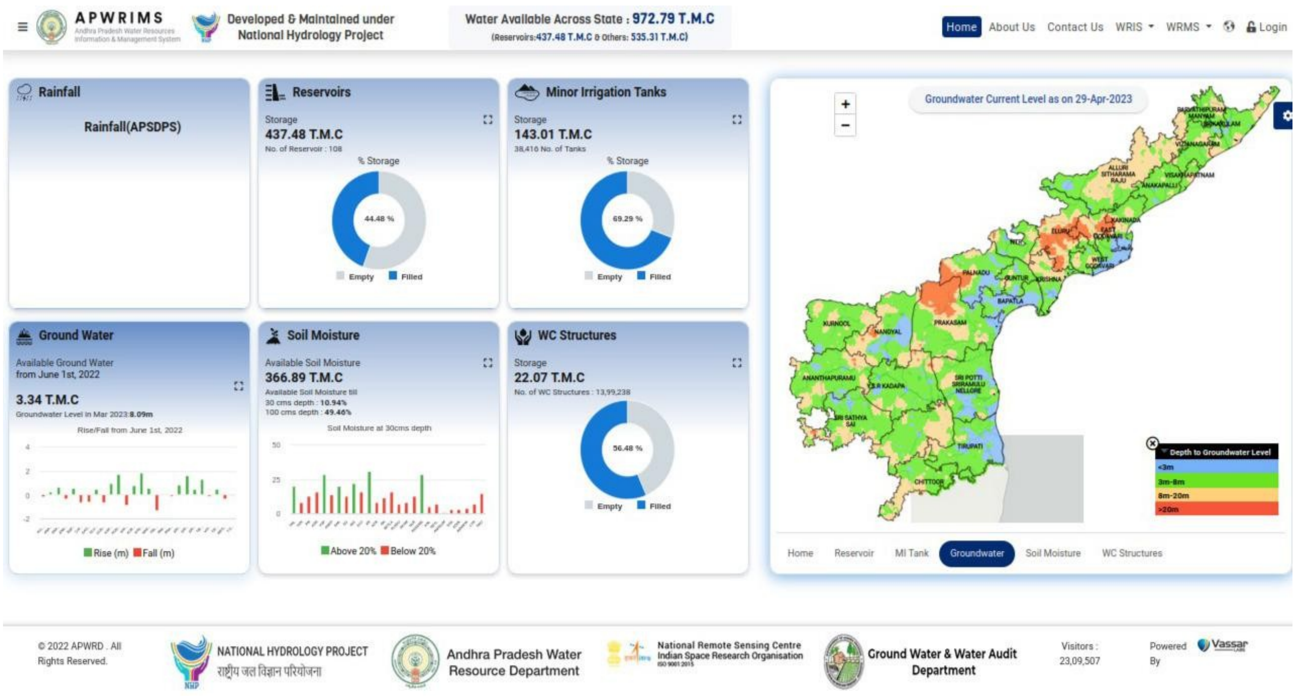
Temporal aggregation can also refer to the process of summarizing data over time periods, such as hourly to daily, monthly, and yearly intervals. This type of aggregation is often used in time-series analysis to identify trends and patterns over time. Hourly data can be aggregated into daily data by summing or averaging the hourly values for each day. Similarly, daily data can be aggregated into monthly data by summing or averaging the daily values for each month. Finally, monthly data can be aggregated into yearly data by summing or averaging the monthly values for each year.

6.1.3 API's

APIs (Application Programming Interfaces) are tools that allow users to access and retrieve information about data resources, such as river gauge station data, rating table data, time-series data, aggregated data etc. They can also to used to update meta data and upload several meta data of river gauge stations, rating table data and historical data.

Overall, APIs are powerful tools that can help users access and utilize software, systems, and data in a more efficient and effective way, while also providing flexibility and customization options.

6.2 Sample Screenshots


APWRIMS — Andhra Pradesh Water Resources Information & Management System

Developed & Maintained under National Hydrology Project

Water Available Across State : 972.79 T.M.C (Reservoirs:437.48 T.M.C & Others: 535.31 T.M.C)

Rainfall: Rainfall(APSDPS)

Reservoirs: Storage 437.48 T.M.C; No. of Reservoir : 108; % Storage 44.48 %

Minor Irrigation Tanks: Storage 143.01 T.M.C; 38,410 No. of Tanks; % Storage 69.29 %

Ground Water: Available Ground Water from June 1st, 2022 3.34 T.M.C; Groundwater Level in Mar 2023:8.09m

Soil Moisture: Available Soil Moisture 366.89 T.M.C; Available Soil Moisture till 30 cms depth : 10.94%; 100 cms depth : 49.46%

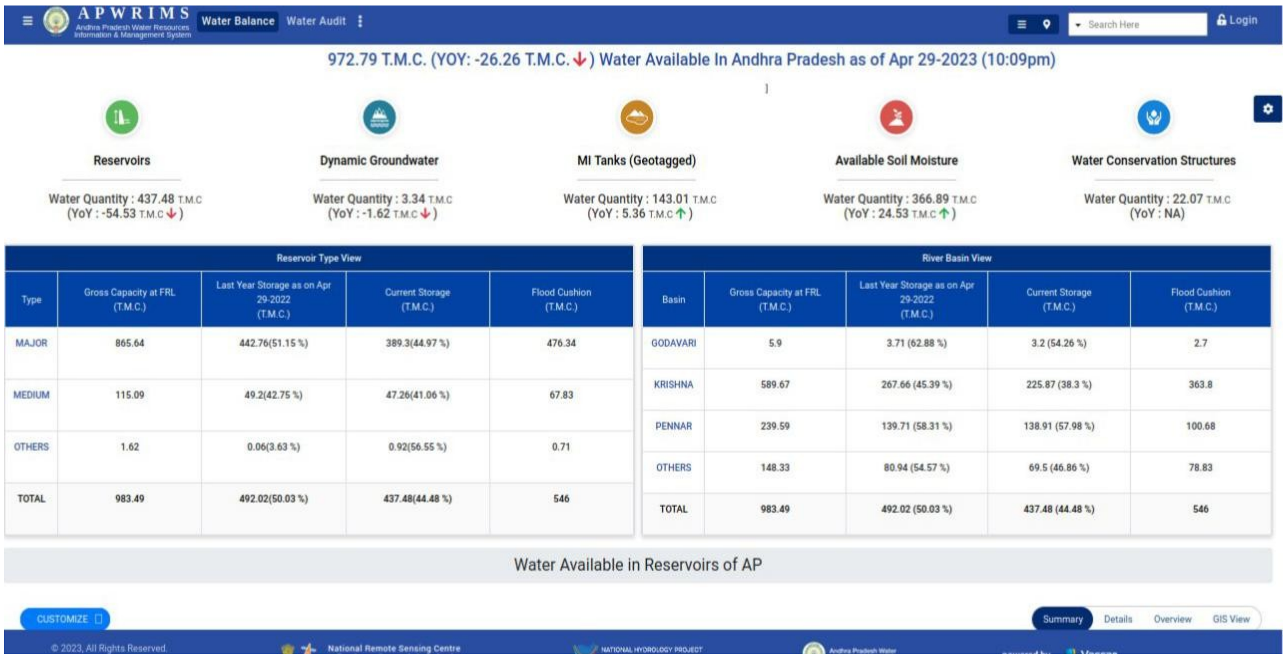
WC Structures: Storage 22.07 T.M.C; No. of WC Structures : 13,99,238; 56.48 %

Groundwater Current Level as on 29-Apr-2023

© 2022 APWRD . All Rights Reserved. Visitors : 23,09,507

Home
About Us
Contact Us
WRIS
WRMS
Login
Empty
Filled
Rise/Fall from June 1st, 2022
Rise (m)
Fall (m)
Soil Moisture at 30cms depth
Above 20%
Below 20%
Depth to Groundwater Level
<3m
3m-8m
8m-20m
>20m
Home
Reservoir
MI Tank
Groundwater
Soil Moisture
WC Structures
NATIONAL HYDROLOGY PROJECT
Andhra Pradesh Water Resource Department
National Remote Sensing Centre Indian Space Research Organisation
Ground Water & Water Audit Department
Powered By Vassar

Figure 6.3.1 Homepage



972.79 T.M.C. (YOY: -26.26 T.M.C.↓) Water Available In Andhra Pradesh as of Apr 29-2023 (10:09pm)

Reservoirs	Dynamic Groundwater	MI Tanks (Geotagged)	Available Soil Moisture	Water Conservation Structures
Water Quantity : 437.48 T.M.C (YoY : -54.53 T.M.C↓)	Water Quantity : 3.34 T.M.C (YoY : -1.62 T.M.C↓)	Water Quantity : 143.01 T.M.C (YoY : 5.36 T.M.C↑)	Water Quantity : 366.89 T.M.C (YoY : 24.53 T.M.C↑)	Water Quantity : 22.07 T.M.C (YoY : NA)

Reservoir Type View

Type	Gross Capacity at FRL (T.M.C.)	Last Year Storage as on Apr 29-2022 (T.M.C.)	Current Storage (T.M.C.)	Flood Cushion (T.M.C.)
MAJOR	865.64	442.76(51.15 %)	389.3(44.97 %)	476.34
MEDIUM	115.09	49.2(42.75 %)	47.26(41.06 %)	67.83
OTHERS	1.62	0.06(3.63 %)	0.92(56.55 %)	0.71
TOTAL	983.49	492.02(50.03 %)	437.48(44.48 %)	546

River Basin View

Basin	Gross Capacity at FRL (T.M.C.)	Last Year Storage as on Apr 29-2022 (T.M.C.)	Current Storage (T.M.C.)	Flood Cushion (T.M.C.)
GODAVARI	5.9	3.71 (62.88 %)	3.2 (54.26 %)	2.7
KRISHNA	589.67	267.66 (45.39 %)	225.87 (38.3 %)	363.8
PENNAR	239.59	139.71 (58.31 %)	138.91 (57.98 %)	100.68
OTHERS	148.33	80.94 (54.57 %)	69.5 (46.86 %)	78.83
TOTAL	983.49	492.02 (50.03 %)	437.48 (44.48 %)	546

Water Available in Reservoirs of AP

CUSTOMIZE
Summary
Details
Overview
GIS View
© 2023, All Rights Reserved.
National Remote Sensing Centre
NATIONAL HYDROLOGY PROJECT

Figure 6.3.2 Water Balance Page

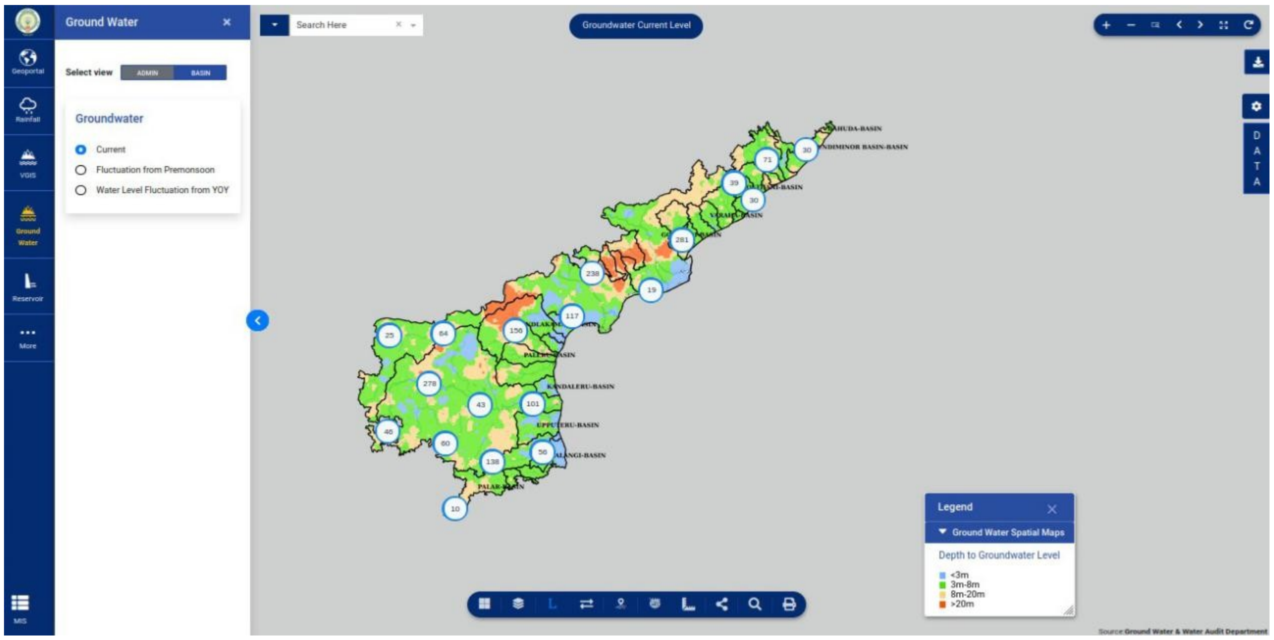

Ground Water
Geoportal
Rainfall
VGIS
Ground Water
Reservoir
More
MIS
Select view
ADMIN
BASIN
Groundwater
Current
Fluctuation from Premonsoon
Water Level Fluctuation from YOY
Search Here
Groundwater Current Level
DATA
Legend
Ground Water Spatial Maps
Depth to Groundwater Level
<3m
3m-8m
8m-20m
>20m
Source:Ground Water & Water Audit Department

Figure 6.3.3 Heat Map Page

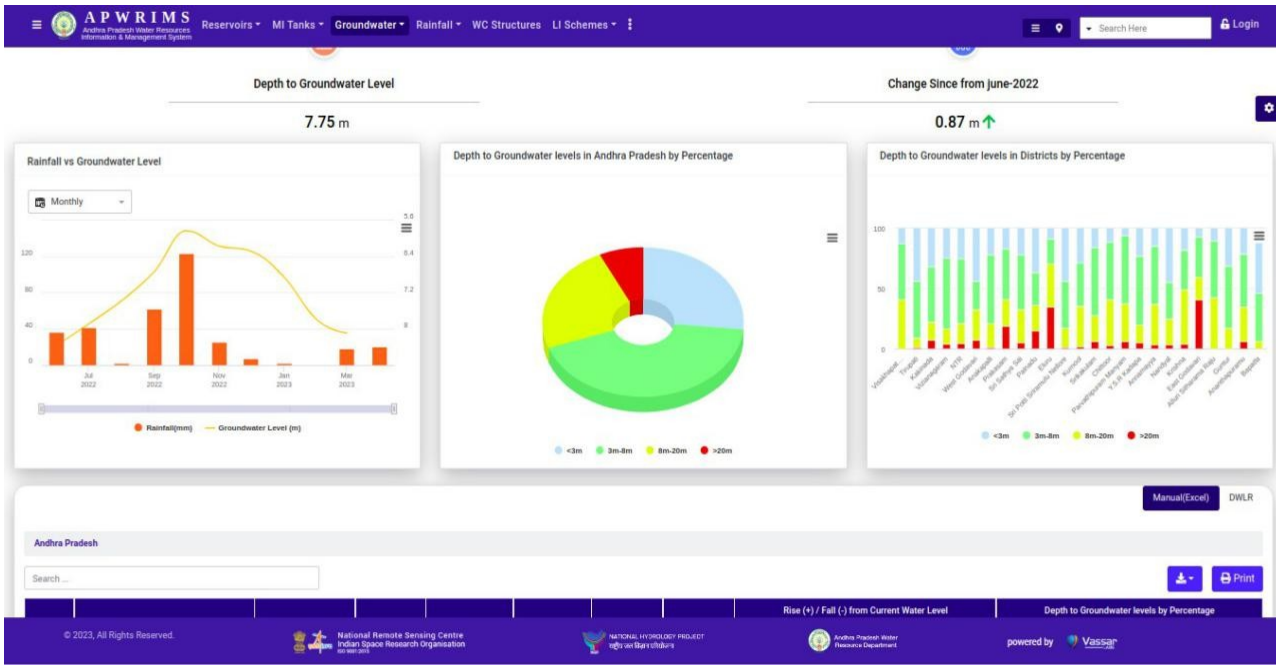

APWRIMS
Andhra Pradesh Water Resources Information & Management System
Reservoirs
MI Tanks
Groundwater
Rainfall
WC Structures
LI Schemes
Search Here
Login
Depth to Groundwater Level
7.75 m
Change Since from june-2022
0.87 m
Rainfall vs Groundwater Level
Monthly
Rainfall(mm)
Groundwater Level (m)
Depth to Groundwater levels in Andhra Pradesh by Percentage
<3m
3m-8m
8m-20m
>20m
Depth to Groundwater levels in Districts by Percentage
Manual(Excel)
DWLR
Andhra Pradesh
Search ...
Print
Rise (+) / Fall (-) from Current Water Level
Depth to Groundwater levels by Percentage
© 2023, All Rights Reserved.
National Remote Sensing Centre
Indian Space Research Organisation
National Hydrology Project
Andhra Pradesh Water Resource Department
powered by
Vassar

Figure 6.3.4 Dashboards Page

CHAPTER 7
TESTING

System tests are designed to validate a fully developed system to assure that it meets its requirements. The test cases are therefore designed solely based on the SRS document. (OR) System testing is nothing but how the customer is going to start using your application and checking whether everything is up to the mark and meeting the needs of the customer.

7.1 Unit Testing:

Unit testing, a testing technique using which individual modules are tested to determine if there are any issues by the developer himself. It is concerned with functional correctness of the standalone modules. Reduces Defects in the Newly developed features or reduces bugs when changing the existing functionality. Improves design and allows better refactoring of code. Unit Tests, when integrated with build gives the quality of the build as well. It is the first level of functional testing. Below are the test cases on the individual modules of the designed website. The functionality of each module has been checked by the developer of the module.

7.1.1 Scraper

This involves creating automated tests that simulate different scenarios, such as successful scraping, connection errors, missing data, and invalid inputs, and verify that the scraped data is in the expected format and matches the data structure used in the system. The tests should ensure that the web scraper code can handle different types of data such as text, images, and links, and that it can handle errors and exceptions gracefully, providing meaningful error **messages.By** conducting unit testing for web scraping, you can catch issues early in the development cycle, ensure that the web scraper code is efficient and compliant with legal and ethical requirements, and improve the overall quality of your software.

7.1.2 Aggregator

This involves creating automated tests that simulate different scenarios, such as successful aggregation, missing data, and invalid inputs, and verify that the aggregated data is in the expected format and matches the data structure used in the system. The tests should ensure that the aggregator code can handle different types of data from different sources, such as sensors and websites, and that it can handle errors and exceptions gracefully, providing meaningful error messages.

7.1.3 API’s

This involves creating automated tests that simulate different scenarios, such as successful API calls, incorrect or missing input parameters, and unexpected server responses, and verify that the API responses are in the expected format and matches the data structure used in the system.

7.2 Integration Testing

Integration testing is the second level of the software testing process comes after unit testing.In this testing,units or individual components of the software are tested in a group.The focus of the integration testing level is to expose defects at the time of interaction between integrated components or units.unit testing uses modules for testing purpose,and these modules are combined and tested in integration testing.

The goal of integration testing is to check the correctness of communication among all the modules.It includes four types of approaches.

A typical software project consists of multiple software modules, coded by different programmers. The purpose of this level of testing is to expose defects in the interaction between these software modules when they are integrated.

Integration testing includes various approaches like

1. Bigbang approach

2. Top down approach

3.Bottom up

approach

4.Sandwiched Integration testing.

For this website, a Sandwiched approach of integration testing has been used. Sandwiched approach is also called a Mixed approach. We performed this testing by collecting individual modules and combining them one by one and tested for their functionalities whether those are working properly or not. Finally, after completion of integration testing, we confirmed that these individual modules worked properly even after being combined with other modules.

CHAPTER 8
CONCLUSION & FUTURE SCOPE

8.1 Conclusion

In conclusion, the project aims to provide real-time insights into various water resource parameters such as rainfall, groundwater level, river water level, inflow forecast, soil moisture, flood forecasting, and reservoir water level. The system collects data from various sensors and websites, processes it using advanced technologies such as Kafka, Flink, Cassandra, and PostgreSQL, and generates dashboards, heat maps, graphs, and tables for visualization. The project uses a wide range of technologies and software tools. These technologies and tools have been chosen carefully to ensure that the system is scalable, reliable, and efficient.

The project addresses several challenges in water resource management such as flood forecasting, reservoir management and planning, canal management system, and water audit and budgeting. The system can help governments and organizations make data-driven decisions to manage water resources effectively.

Overall, the project has the potential to bring significant benefits to the water resource management sector by providing real-time insights, optimizing resource utilization, and improving water security. The project also has future scope for further development and expansion, such as integrating with other systems and adding new features to address emerging challenges in the field of water resource management.

8.2 Future Scope

The proposed system has significant potential for future expansion, including the integration of additional sensors and data sources, the development of new DSS products, and the deployment of the system in other regions.

Here are some possible areas that can be explored further:

- Additional water parameters: The project can be further developed to include additional water parameters, such as water quality, sediment levels, and dissolved oxygen levels. This will provide a more comprehensive view of the water resource and help in better decision-making.

- Integration with other systems: The project can be integrated with other systems, such as weather forecasting systems and agricultural data systems. This will provide a more complete view of the factors affecting water resources and help in better planning and management.
- Improved analytics: The project can be further developed to include more advanced analytics, such as predictive analytics and machine learning algorithms. This will enable better forecasting and decision-making and help in identifying trends and patterns that are not easily visible with traditional methods.
- User customization: The project can be further developed to allow users to customize the dashboard and visualizations according to their specific needs. This will make the system more user-friendly and enable users to get the insights they need quickly and easily.
- Increased automation: The project can be further developed to increase automation, such as automated data collection and analysis, to reduce human error and make the system more efficient.
- Expansion to other regions: The project can be expanded to cover other regions and countries, which will increase the scope of the project and enable better global water management.
- Enhanced reporting: The project can be further developed to include more advanced reporting features, such as custom reports and data exports. This will enable users to generate reports quickly and easily and share the data with other stakeholders.

CHAPTER 9

REFERENCES

[1] https://www.stackoverflo w.com/

[2] https://cassandra.apache.org/doc/latest/

[3] https://www.postgresql.org/docs/

[4] https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/

[5] https://kafka.apache.org/documentation/

[6] http://www.quartz-scheduler.org/documentation/

[7] https://nightlies.apache.org/flink/flink-docs-master/

[8] https://docs.github.com/en

[9] https://docs.oracle.com/en/java/javase/11/docs/api/