Project Report

On

EPINET User Account Creation Request Authorization System

Submitted for the partial fulfilment of the requirement for

the completion of

Summer Training/Internship

at

ONGC Academy, KDMIPE, Dehradun

Compiled by:

Ruudra Amola

4th year Student, B. Tech CSE

Graphic Era Deemed to be University, Dehradun

Project Mentor:

Mr. Subodh Kumar

General Manager (Programming)

EPINET, ONGC, Dehradun

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to **Oil and Natural Gas Corporation Limited, Dehradun** for providing me with the opportunity to undertake my summer training as a part of my academic curriculum. This internship has been an invaluable experience, allowing me to gain practical insights into the field of Web Development and its Applications and apply the theoretical knowledge acquired during my studies.

I extend my heartfelt thanks to **Mr. Subodh Kumar, GM(Prog.)** at EPINET Data Center for their guidance and support throughout the internship period. Their expertise played a pivotal role in shaping my understanding of the industry and enhancing my technical skills.

I am also thankful to the entire team at EPINET and especially **Mr. DK Dhruw**, **GM(Prog.), EPINET** for his warm welcome and cooperation during my time with the organization. His willingness to share knowledge and calm, supportive nature greatly contributed to my learning process during that time.

Lastly, I am indebted to my institution **Graphic Era Deemed to be University, Dehradun** for facilitating this internship opportunity and providing me with the necessary academic foundation to make the most of this experience. I am confident that the insights gained during this internship will serve as a strong foundation for my future endeavors.

Ruudra Amola
Final year, B. Tech CSE
Graphic Era Deemed University, Dehradun

CONTENTS

Introduction about ONGC

Oil and Natural Gas Corporation Limited (ONGC) is a prominent Indian multinational oil and gas exploration and production company. Established in 1956, ONGC holds a significant position in the energy sector and plays a vital role in India's quest for self-sufficiency in oil and gas production. The company is entrusted with the exploration, development, and production of crude oil and natural gas, both offshore and onshore.

With its headquarters located in Dehradun, India, ONGC operates across various regions, both within the country and internationally. ONGC's operations encompass a wide spectrum of activities, including exploration for hydrocarbons, drilling of wells, production of oil and gas, and the development of related technologies.

ONGC's contributions extend beyond the realm of energy production. The company also emphasizes sustainable practices, environmental conservation, and community development through its corporate social responsibility initiatives. ONGC's commitment to innovation, technological advancement, and responsible business practices has earned it a prominent position in India's industrial landscape.

Frontiers of Technology at ONGC

- **Advanced Reservoir Insights**: ONGC pioneers 3D/4D seismic and simulation for precise reservoir understanding and optimized hydrocarbon extraction.
- **Advanced Drilling Techniques**: ONGC leads with directional drilling, extending reach techniques, and managed pressure drilling for efficient exploration.
- **Data-Driven Operations**: ONGC employs IoT and data analytics to enhance safety, minimize downtime, and optimize resource management.
- **Enhanced Oil Recovery (EOR) Techniques**: ONGC explores polymer flooding, chemical EOR, and more to boost recovery from mature oil fields.
- **Green and Sustainable Solutions**: ONGC embraces carbon capture, renewables integration, and waste heat recovery for eco-friendly operations.

Introduction about EPINET Data Centre

Over the past few years E&P Information technology has incrementally made significant strides in the workflow process dynamics of upstream petroleum industry. There is an increasing desire and thrust for long-term preservation of data assets by building up corporate E&P information and knowledge networks to seek business decisions. Meanwhile the corporate world has begun to recognize that data is a high asset. Internet, Intranet, and web-based technologies are promising a single interface across multiple computer platforms. thereby promising better, cross discipline, concurrent usage of data & therefore EPINET project was set up.

Exploration & Production Information Network (EPINET) has emerged as one source of information by collecting the data of different activities i.e., Seismic, drilling, well, logging & production for achieving one goal of the organization for development of an E&P company like ONGC. The project is intended to establish an organization wide dynamic database having GIS features and Web capabilities, to loosely interconnect different data stores located at geographically diversified locations.

1. The project EPINET Phase-I was initiated & installed IT infrastructure like computer systems and data management software were acquired under LAN/WAN environment at KDMIPE, Dehradun, Baroda, Chennai, Mumbai, Nazira, and Ahmedabad.

2. Phase - II of the project is designed to ensure complete execution and successful operationalization. ONGC carried out a comprehensive work-study under Phase-I to identify Phase-II job elements and the recommendations were further validated.

3. In Phase-III installation of Hardware & Software (H/W & S/W) at Assets/Basins/Forward Bases completed and migration of legacy Exploration & Production (E&P) data (Managed only physical asset metadata in phase – II at Dehradun) into Exploration & Production Information Network (EPINET) system at basins, assets and other work centers started. The data types given emphasis in Phase-II are drilling, Seismic trace data, production, and reservoir along with the remaining Phase-I data.

EPINET IT infrastructure

Hardware: Sun Microsystems

Operating System (OS): Sun Solaris 8

Database Management: RDBMS Oracle 9i System (DBMS)

E&P DM Technology: Finder Data Management System, Log DB, Seismic DB

EPINET Data Types

Well Data: General Well data reported in well completion reports (i.e., core details, stratigraphy, well completion, production testing, reservoir, rock / fluid sample analysis results etc.) pertaining to over 10,000 wells are being managed in Finder master database.

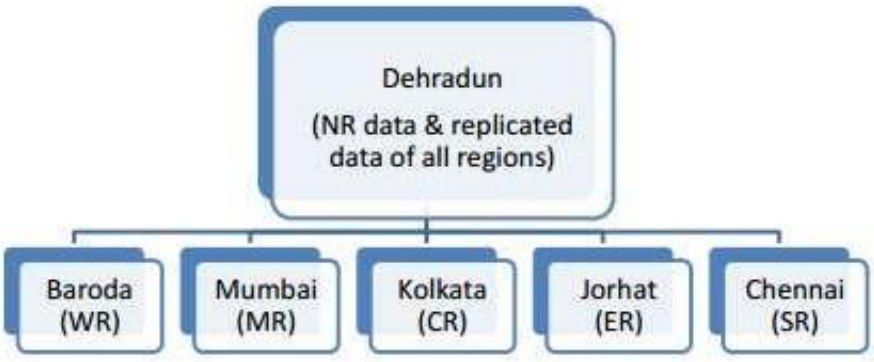
Log Data: ONGC has accumulated a tremendous amount of log data in its 50 years of operation. The number of well log tapes reached thousands, storing hundreds of thousands of well log curves. ONGC manages raw logs as well as spliced, edited, and ready to use merged logs by utilizing Log DB and Finder master database technologies.

Seismic Data: Most of the seismic navigation data was managed during Phase-I and post Phase-I activities. However, the entire 2D/ 3D seismic stack and migrated data was covered in Phase-II and managed by utilizing Seis DB software from the solution provider.

Database Centres & Data Flow

Under ProSource, EPINET was organized into six data centres in a two-tier model with the corporate centre housed at KDMIPE, Dehradun. Data from all centres are centrally replicated at the Corporate EPINET Centre, Dehradun via Golden Gate, a plugin of Oracle. The data centres and the regions covered are as follows:

Data Center Location	Basin / Region covered	Region
Dehradun	Frontier Basin	Northern Region (NR)
Baroda	Western Onshore Basin	Western Region (WR)
Mumbai	Western Offshore	Mumbai Region (MR)
Kolkata	MBA Basin	Central Region (CR)
Jorhat	A&AA Basin	Eastern Region (ER)
Chennai	KG-PG Basin	Southern Region (SR)



EPINET User Account Creation Request Authorization System

Abstract and Idea

ONGC has a large amount of data stored in SQL Databases. EPINET data centre is basically a data repository where data reports of log, well and seismic data are stored. To access these reports any employee needs to have an account in the EPINET portal and for that a complete request process must be completed. The employee must fill in a form that goes through a 3-level authentication course by the Controlling Officer, the Head of the Department and then the Database Administrator. This process is performed at a physical level currently.

Objective

The objective of this project is to create a prototype for the EPINET user account creation request system which also allows authorization of the request at the 3 levels.

Scope of the Project

The scope of the project entails the development of a prototype for the EPINET user account creation request system, integrating a streamlined process for authorization at three distinct levels. The current manual process involves the submission of a form by an employee who wishes to access data reports stored in the EPINET data centre. The three levels of authentication involve the Controlling Officer, the Head of the Department, and the Database Administrator.

This is just a closest prototype of the potential system which uses a local database created using XAMPP Control panel and PHP scripts. I have used a local database as access to the original EPINET database is not available to me.

PROJECT REQIREMENTS

S. No	Software Requirements	
1.	Operating System	Windows 7,8,10,11 / Linux
2.	Platforms used	VS Code, XAMPP Control Panel, PHPMyAdmin
3.	Languages used	SQL, HTML5, CSS3, JavaScript, PHP
4.	Technologies used	Full Stack Web Development

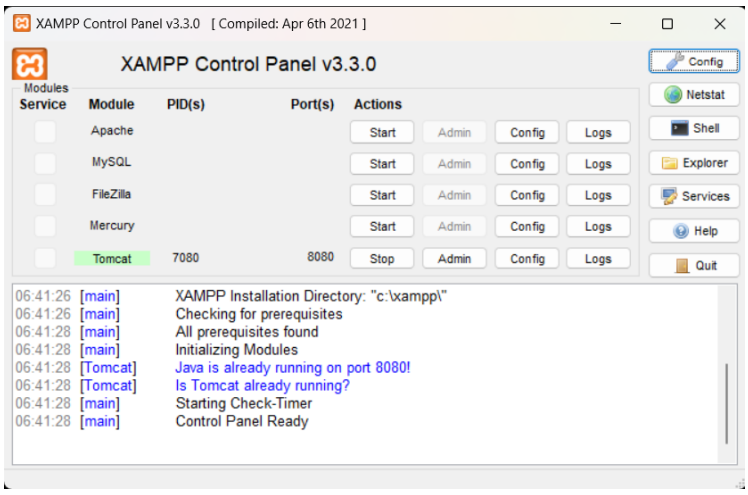
Platforms Used

Visual Studio Code (version 1.81) -

Visual Studio Code (VS Code) is a widely used source code editor developed by Microsoft. It's designed to be lightweight, highly customizable, and efficient for various programming languages and development tasks. VS Code provides features like syntax highlighting, autocompletion, debugging, version control integration, and an extensive collection of extensions that enhance its functionality. Its intuitive user interface and powerful capabilities make it a popular choice among developers for writing, editing, and managing code efficiently.

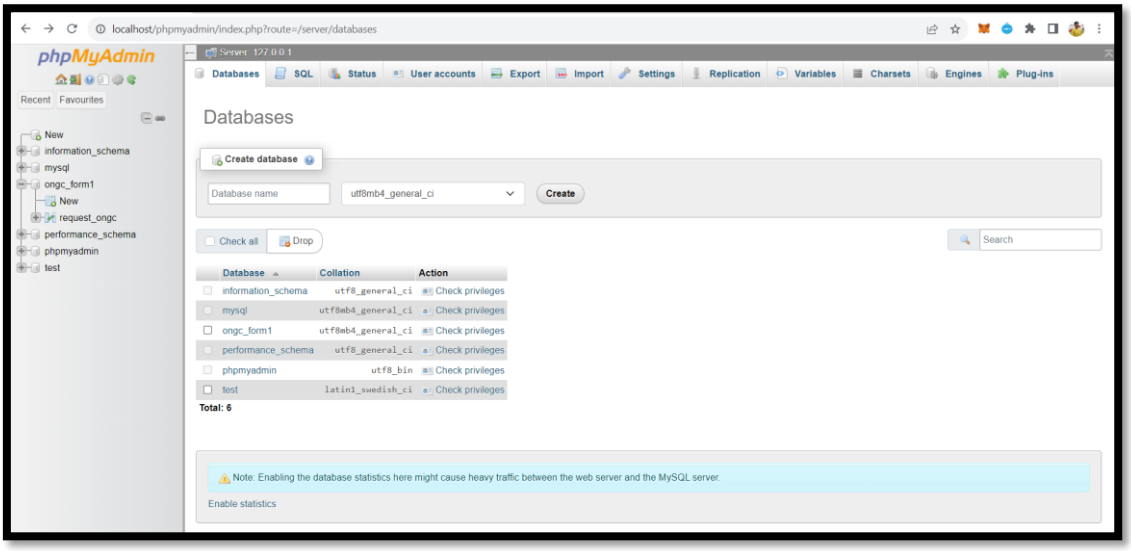
XAMPP Control Panel -

XAMPP Control Panel is a software package that simplifies the setup and management of a local web server environment. XAMPP stands for "Cross-Platform, Apache, MySQL, PHP, and Perl," indicating the core components it includes. The Control Panel allows you to easily install and configure these components, along with other tools like PhpMyAdmin, on your local machine. This local server environment provides a user-friendly interface to manipulate server, manage databases, and access other tools necessary for building and testing web projects.



PHPMyAdmin Portal –

phpMyAdmin is a web-based tool that simplifies the management of MySQL and MariaDB databases. When you use XAMPP, phpMyAdmin is included as one of the tools you can access from the XAMPP Control Panel. This means you can easily manage your local databases without needing to set up a separate interface or connect through the CLI. One can create, modify, and delete databases, tables, and fields, as well as import and export data in various formats. It also offers tools for executing SQL queries, managing user privileges, and generating reports about the database structure and content.



Languages and Scripts Used

Structured Query Language (SQL) –

SQL is a programming language designed for managing and manipulating relational databases. It's used for tasks like creating, querying, updating, and deleting data within databases. SQL allows you to interact with databases to retrieve specific information, add new data, modify existing data, and perform various administrative tasks.

Hypertext Markup Language (HTML) –

HTML is the standard markup language used to create and structure content on the World Wide Web. It consists of a set of elements and tags that define the structure and presentation of web documents, such as websites and web pages. HTML elements are used to define headings, paragraphs, links, images, forms, and other elements that make up the content of a web page.

Cascading Style Sheet (CSS) –

CSS is a stylesheet language used to define the visual presentation and layout of HTML and XML documents, including web pages. It provides a way to separate the content (defined using HTML) from its visual appearance and formatting. With CSS, you can control various aspects of how a web page looks, such as colors, typography, layout, positioning, animations, and more.

JavaScript –

JavaScript is a versatile and widely used programming language primarily known for its role in adding interactivity and dynamic behavior to web pages. It is a client-side scripting language that runs in web browsers and enables developers to create interactive user interfaces and manipulate web page content without requiring a full page reload. Key features and uses of JavaScript include Client-Side Interactivity, DOM Manipulation, Event Handling, Web APIs, Frameworks and Libraries, Data Manipulation, Dynamic Styling.

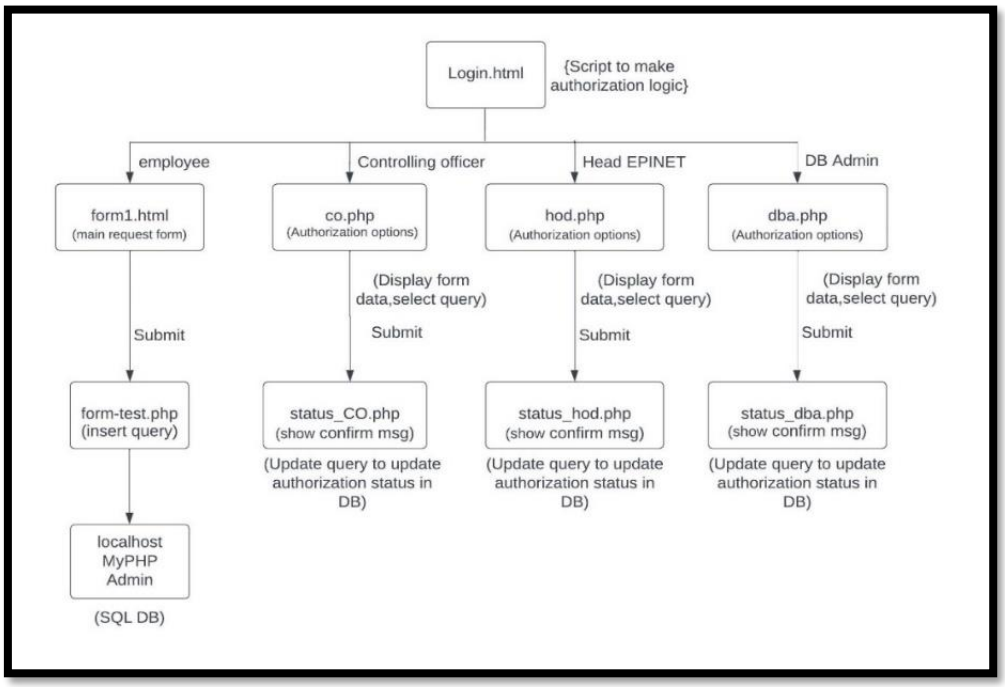
PHP Script –

Hypertext Preprocessor (PHP) is a server-side scripting language designed for backend web development. It's used to create dynamic and interactive web pages by embedding code within HTML or without it. PHP can retrieve data from databases, generate content based on user input, and perform various server-side tasks, enabling the creation of feature-rich web applications. It enables the developer to work with the backend side of the website.

METHADOLOGY

Basic Workflow of pages

The whole project comprises of multiple sets of HTML and PHP files which are linked together in a way to maintain a systematic workflow and connection with the database. The following flowchart shows the flow of control of the whole system.



- The login page i.e., **login.html** holds the structure of the login page. A JavaScript file is embedded in it that holds the login authentication logic.

- The main request form that is to be filed by a person of interest is structured in the form1.html file.

- The form is linked to the CSS file **form1.css** and a JavaScript to include client-side validation for the fields using form1.js file.

- The **form-test.php** is the script file that includes the SQL query to insert the data into the MySQL database using phpMyAdmin.

- The login page redirects to the respective pages with the corresponding login credentials of the user.

- Script files **co.php, hod.php, dba.php** display the data of the latest form request that has been submitted using an SQL query and include options for each person to validate the form data.

- These 3 files are linked to the same CSS file called **display.css** for basic styling features to the page.

- Script files **status-co.php, status-hod.php, status-dba.php** displays the confirmation message that the authorization response has been recorded and runs an SQL query to update the form status in the concerned column of the database at the backend.

- These 3 files are linked to the same CSS file called **status.css** for basic styling features to the page.

- The login page is linked to the login**_style.css** file to include basic page styling.

- Every page except the login one, has a logout option that redirects the user to the login page only.

Detailed workflow with Code Snippets

- The system starts with a login page that consists of basic page structure with username and password fields. A JavaScript is included to prepare a login page authorization logic.

```
loginForm.addEventListener('submit', function (event) {
    event.preventDefault();

    const username = document.getElementById('username').value;
    const password = document.getElementById('password').value;

    const predefinedUsers = {
        'user1': 'user1',
        'ControllingOfficer': 'ControllingOfficer',
        'HeadOfDept': 'HeadOfDept',
        'DBAdmin': 'DBAdmin'
    };
```

- As soon as the login credentials are entered and the login is submitted, the 'submit' event is grabbed by an event listener.

- The username and password values that are entered are read into the variable's **'username'** and **'password'**. This is done using JavaScript DOM manipulation function **getElementById ()**.

DOM

DOM stands for **"Document Object Model"**. It is an API provided by web browsers that allows scripts (like JavaScript) to interact with the structure, style, and content of web documents. When a web page is loaded into a browser, the browser parses the HTML or XML code and creates a tree-like structure of objects that represent the various elements and their relationships within the document. This structure is known as Document Object Model.

- The JavaScript object **'predefinedUsers'** is a variable that holds a JSON set of key-value pairs.

- The login page does not have an option to register first as the system is designed for internal use of the organization and only the existing employees can access it.

- For that reason, anyone who logs in to the system must have his/her credentials updated in the 'predefinedUsers' object.

- The following code runs the conditional logic for user credential authorization and which credential when entered redirects to which page based on privileges.

```
if (predefinedUsers.hasOwnProperty(username) && predefinedUsers[username] === password) {
    alert('Login successful! Redirecting...');

    if (username==='ControllingOfficer'){
        window.location.href = 'http://localhost/Intern proj logins/co.php';
    }
    else if (username==='HeadOfDept'){
        window.location.href = 'http://localhost/Intern proj logins/hod.php';
    }
    else if (username==='DBAdmin'){
        window.location.href = 'http://localhost/Intern proj logins/dba.php';
    }
    else{
        window.location.href = 'http://localhost/Intern proj logins/form1.html';
    }
```

- The outer if condition checks that if the entered credentials are present in the **'predefinedUsers'** object then the control enters the loop otherwise a pop up will show "Invalid credentials" message.

- There are 3 main privileges that are to be granted to 3 people i.e., the Controlling Officer then The Head of the Department or the Head EPINET and the Database Administrator at the last.

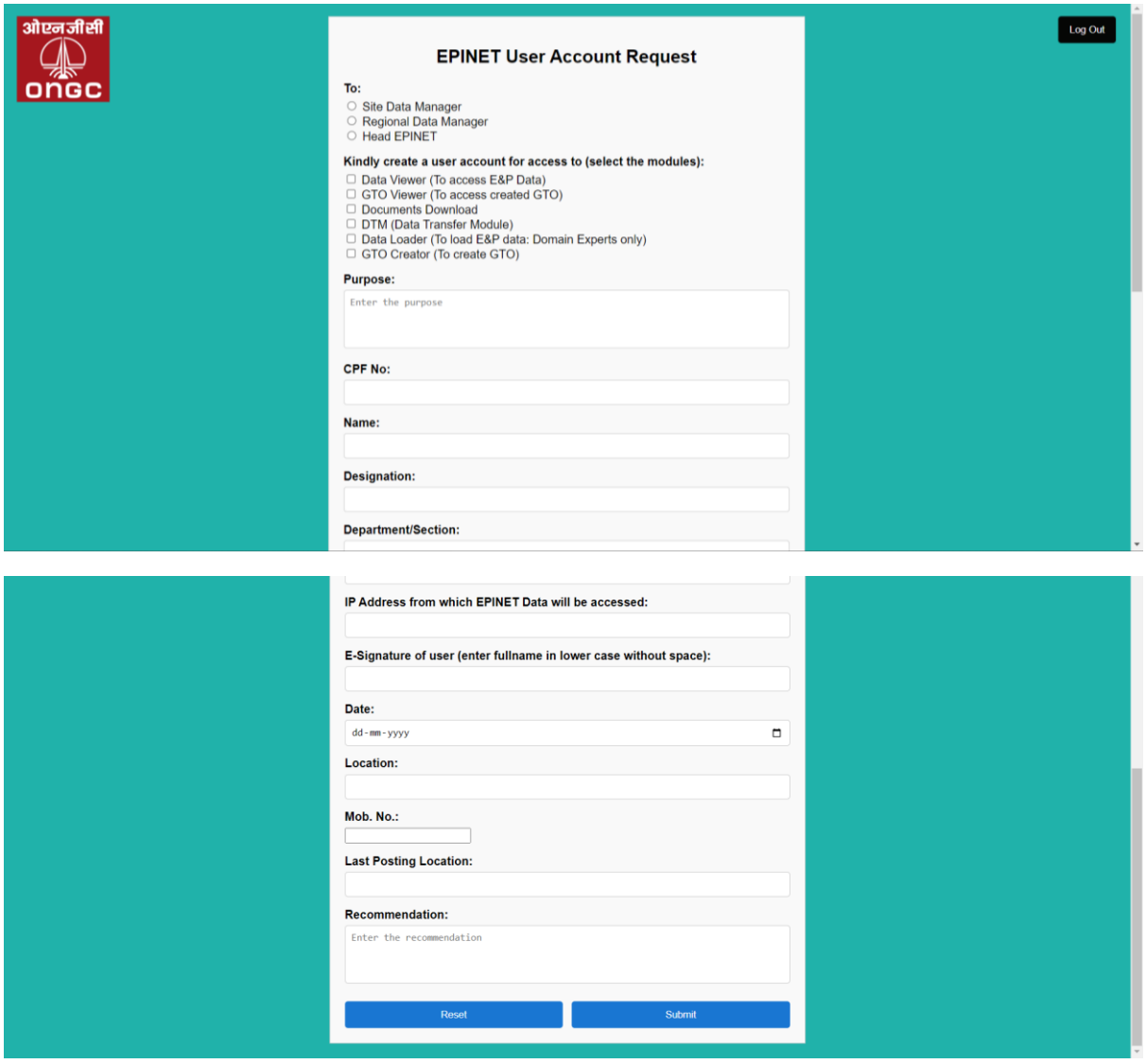
- For these 3 people 3 if conditions are written that redirects them to their respective pages where they can review the latest request and give their authorization for the same.

- But the last else condition is for every employee other than the controlling officer, head EPINET and the DBA. Any other person who logs in as a normal employee will be who can only fill in the request form and cannot authorize any data.

- This will redirect the user to the main request form in the file **'form1.html'**.

- The HTML and CSS code for the following EPINET User Account creation Request form is a very lengthy code, so the complete code is available in the GitHub repository link in the file named **'form1.html'** that I have mentioned in the bibliography section at the end of this report.

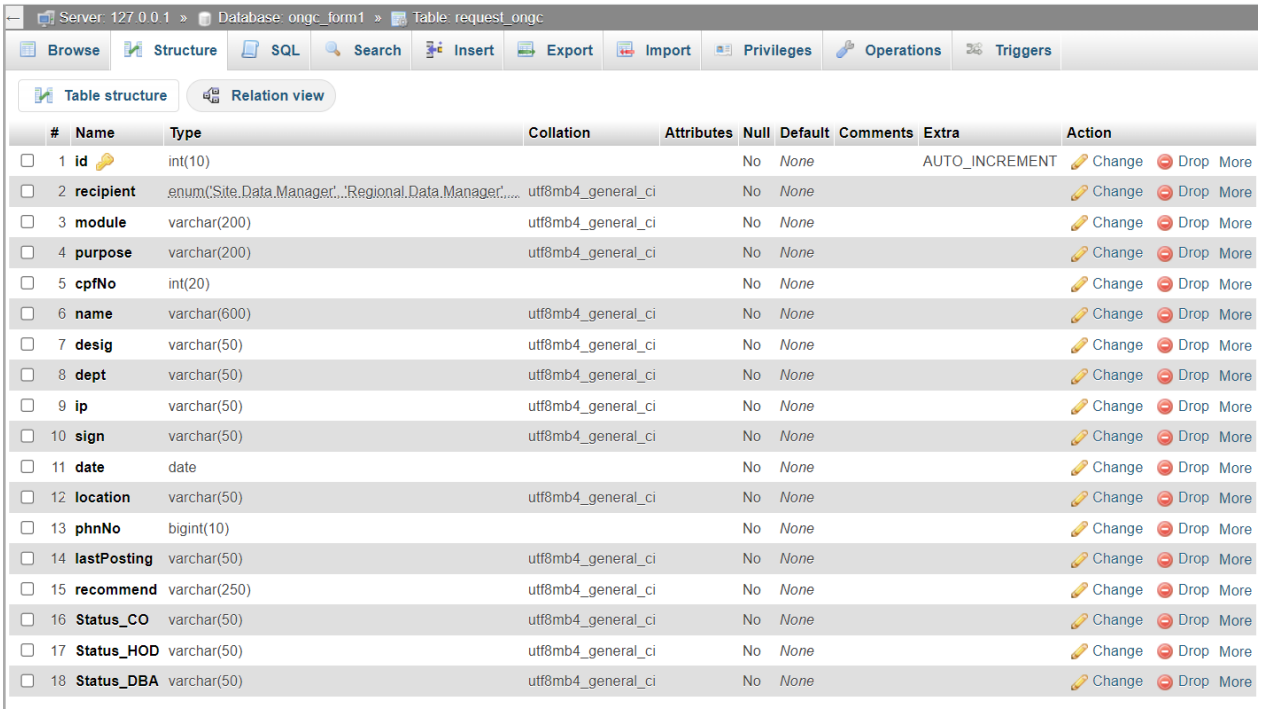
form1.html



- As an example, for this prototype system I have included a credential in the 'predefinedUsers' JSON object as **'user1' : 'user1'** as a username-password pair that denotes a normal employee who when logs in will be presented with this form.

- When the user submits this form, the data will be sent to the local MySQL database that is available in phpMyAdmin using PHP file **'form-test.php'** that runs the insert query in the backend.

- The following is the structure of the table named **'request_ongc'** in the database named **'ongc_form1'**. This is a relational database that is based on MySQL logic.



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(10)			No	None		AUTO_INCREMENT	Change Drop More
2	recipient	enum('Site Data Manager', 'Regional Data Manager',...	utf8mb4_general_ci		No	None			Change Drop More
3	module	varchar(200)	utf8mb4_general_ci		No	None			Change Drop More
4	purpose	varchar(200)	utf8mb4_general_ci		No	None			Change Drop More
5	cpfNo	int(20)			No	None			Change Drop More
6	name	varchar(600)	utf8mb4_general_ci		No	None			Change Drop More
7	desig	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
8	dept	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
9	ip	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
10	sign	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
11	date	date			No	None			Change Drop More
12	location	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
13	phnNo	bigint(10)			No	None			Change Drop More
14	lastPosting	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
15	recommend	varchar(250)	utf8mb4_general_ci		No	None			Change Drop More
16	Status_CO	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
17	Status_HOD	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
18	Status_DBA	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More

- The table has 18 fields out of which 17 are from the form and the first column is the extra one named **'id'** which is the auto incremental primary key in the table.

- Column number 16, 17 and 18 are the main columns that store the status of each of the three authorization levels. The default value into the 3 columns at the time of insertion is **"Pending"**.

- And as soon as any of the 3 concerned authorities reviews the submitted form data and selects his validation status of the form, the respective status field is updated in the database.

- In the following form entry, it is visible that the columns **Status_HOD** and **Status_DBA** is 'Pending' as The Head EPINET and the DBA have not yet reviewed the form data.

id	recipient	module	purpose	cpfNo	name	desig	dept	ip	sign	date	location	phnNo	lastPosting	recommend	Status_CO	Status_HOD	Status_DBA
31	Head EPINET	DTM (Data Transfer Module), Data Loader	testing purposes	500123	Aman Gupta	Employee	EPINET	127.001.500	amangupta	2023-08-16	Dehradun	9319231103	Delhi	Recommendation is good.	Under Review	Pending	Pending

- When the user submits the request form and the form passes its validation process, a PHP file in the backend is called which runs an SQL query to insert the form data into the database.
  Here, **'form-test.php'** is the file with the PHP script written inside HTML tags to insert the form data into the database table.

form-test.php

```php
<?php
    $servername = "localhost";
    $username = "root";
    $password = "";
    $dbname = "ongc_form1";

    $conn = new mysqli($servername, $username, $password, $dbname);
```

- In the above code snippet, a localhost server is used to establish the database connection with the database called **'ongc_form1'.** The variable **$conn** is the connection object with the MySQL database and is accessed using the XAMPP control panel.

```php
$query_in = "INSERT INTO request_ongc VALUES (
    '$id','$recipient','$data','$purpose','$cpfNo','$name','$desig','$dept',
    '$ip','$sign','$date','$location','$phnNo','$lastPosting','$recommend',
    'Pending','Pending','Pending')";

if(mysqli_query($conn, $query_in)){
    echo "<script> alert('Data stored successfully.') </script>";
} else{
    echo "<script> alert('Data upload unsuccessful.') </script>";
}
mysqli_close($conn);
```

- In the above code, an insertion query is initiated into the variable **$query_in** which inserts the form data that has been stored in the variables like **$recipient, $data, $name, $desig** and so on.

- These values are stored in the variables using the **$_POST** super global variable in PHP. **$_POST** is used to send the relevant data from the client side over to the server.

- Then there is a simple if-else block which checks if the query has been executed to the database or not. In either case it throws a confirmation through an alert pop-up. At the end the connection is closed.

- Also, a confirmation message on the UI of the webpage is displayed which says that the user response has been recorded, along with a logout button to the login home page.

Authorization process

- The next stage of the system is about authorization of the submitted form request by the three concerned authorities i.e., the Controlling officer, the Head EPINET/SDM/RDM and then the Database Administrator.

- All the three people when they log in through their respective credentials will be displayed the same form data that has been uploaded at the latest along with the present authorization for each of the 3 levels.

- Suppose the Head EPINET logs in, he will be able to see all the form data and the current validation status with respect to the other 2 authorities like whether they have **rejected** the request or **approved** it or held it for **review**.

- In case no one has updated their status yet. The status will show a **'Pending'** message to the user.

IMPORTANT NOTE: When the controlling officer, the Head EPINET and the DBA logs in with their user credentials, they are directed to 3 different webpages for each person.

The controlling officer, Head EPINET and the DBA are redirected to **co.php, hod.php** and **dba.php** respectively. Each of these pages display the same content but the only difference is that each page updates the status for the concerned authority.

Once the form status is selected by the user on these pages, each of these pages, **co.php, hod.php** and **dba.php** will redirect to 3 other webpages for each that runs the SQL query to update the status in the database.

Each will redirect to **status_CO.php, status_hod.php,** and **status_dba.php** respectively. And all these 3 files hold the same query but to update 3 different columns in the database.

- Again, a MySQL database connection is initiated using the PHP script.

co.php / hod.php / dba.php

```php
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
  die("Connection failed: " . $conn->connect_error);
}

$sql = "SELECT * FROM `request_ongc` ORDER BY id DESC LIMIT 1;";
$result = $conn->query($sql);
```

- This SQL query is written and stored in the variable **$sql** and the result of the query is stored in the variable **$result.**
- This SQL query retrieves all columns of data from the **'request_ongc'** table, orders the rows based on the **id** column in descending order, and then limits the result set to only the **latest record** (row with the highest id).

```php
if ($result->num_rows > 0) {···
  }
    echo "</ul>";
  }
  else {
    echo "NO LATEST SUBMISSION TO DISPLAY!";
  }
```

- It checks if the database query result has more than 0 rows **($result->num_rows > 0)**.
- If there are rows in the result, it enters the if block to process and display the data. In case there are no rows, it displays a message indicating that there are no submissions to display.

```php
while ($row = mysqli_fetch_assoc($result)) {
  echo "<li>";
  echo "<strong>Recipient :</strong> " . $row['recipient'] . "<br><br>";
  echo "</li>";
  echo "<li>";
  echo "<strong>Modules : </strong> " . $row['module'] . "<br>";
  echo "</li>";
  echo "<li>";
  echo "<strong>Purpose : </strong> " . $row['purpose'] . "<br>";
  echo "</li>";
```

- Inside the **if block**, a loop is used to iterate through each row of the query result **(while ($row = mysqli_fetch_assoc($result))**. For each row of data in the result, it displays the relevant form data they belong to.

- In short, this PHP code fetches data from a database query result and formats it into an HTML list.

```html
<form action="status_CO.php" method="post" class="center" id="statusForm">
<label id="val"><strong>Validate as Divisional Head/Controlling Officer : </strong></label>
          <div class="radio-options">
                  <input type="radio" name="status" value="APPROVED" id="radio1">
                  <label for="radio1">Approve</label>

                  <input type="radio" name="status" value="REJECTED" id="radio2">
                  <label for="radio2">Reject</label>

                  <input type="radio" name="status" value="Under Review" id="radio3">
                  <label for="radio3">Review</label>
          </div>
          <div class="form-buttons">
            <button class="button2" type="reset">Reset</button>
            <button class="button2" type="submit">Submit</button>
          </div>
</form>
```

- In summary, this form is a part of **'co.php'** script only and is used to collect the status decision from **a Divisional Head or Controlling Officer** regarding the form data. The user can choose one of the provided options **("Approve," "Reject," or "Review")** using radio buttons.

- After making a choice, the user can either reset the form or submit their decision. The submitted form data will be sent to the **"status_CO.php"** script for handling, where further actions can be taken based on the selected status.

- Next up the control goes to **"status_CO.php"** script which runs an SQL query in the backend to update the Controlling officer's response regarding the form onto the database.

status_CO.php / status_hod.php / status_dba.php

```
$sql2 = "SELECT * FROM `request_ongc` ORDER BY id DESC LIMIT 1";
$result2 = $conn->query($sql2);
while ($row = mysqli_fetch_assoc($result2)){
    $id = $row['id'];
}

$sql = "UPDATE request_ongc SET Status_CO='$status_CO' WHERE id='$id'";
$result = $conn->query($sql);
```

- The SQL query **$sql2** is used to retrieve the id of the latest entry in the database for which the Controlling officer's status will be updated. And then it is used in the updation query next.

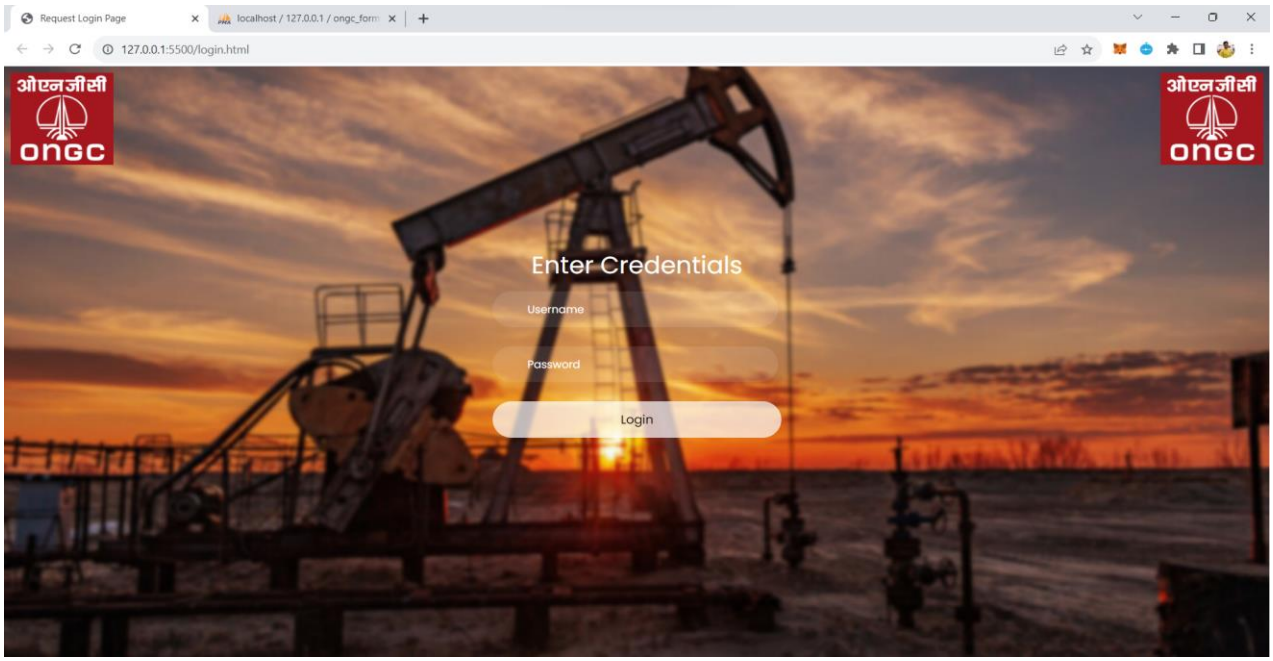
- The SQL query **$sql** updates the **Status_CO** column in the **'request_ongc'** table for the records that match the specified id value. The new value for the **Status_CO** column is taken from the **$status_CO** variable or parameter. The query allows you to update the status of a specific record identified by its id.

A similar process flow will take place to update the validation status by the Head EPINET and then the Database Administrator.

RESULTS AND EXPLANATIONS

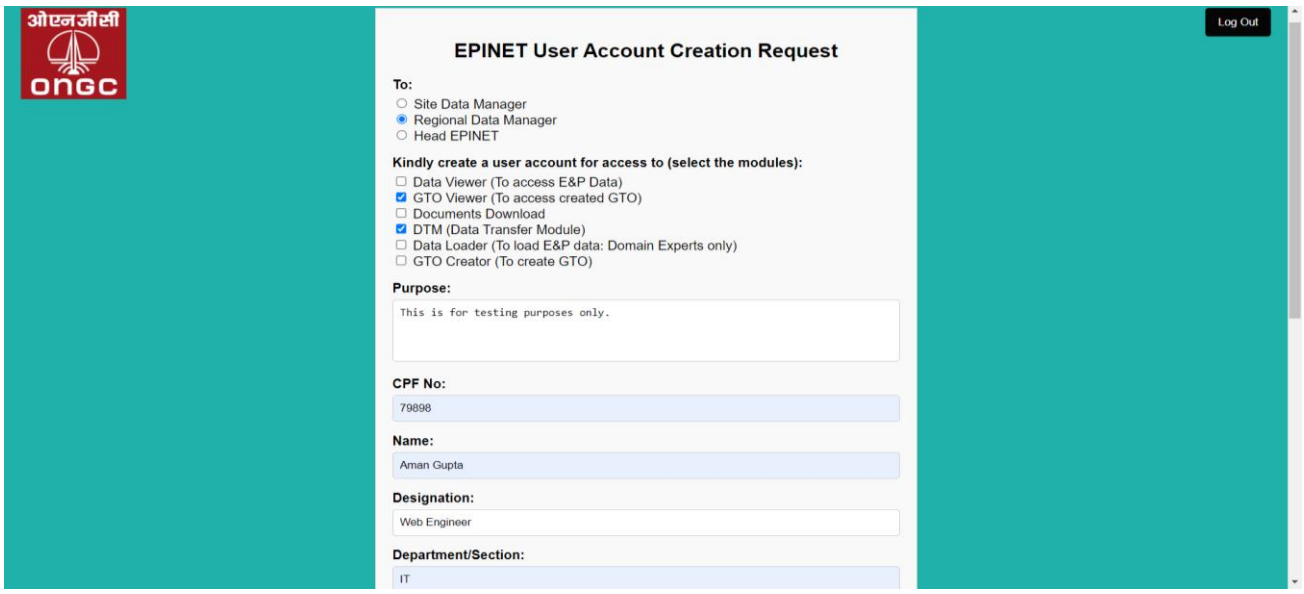
This section includes the snapshots of the webpages in order of the systematic flow of the project. It starts with the login home page till the point where all 3 of the authorization status are updated.

login.html

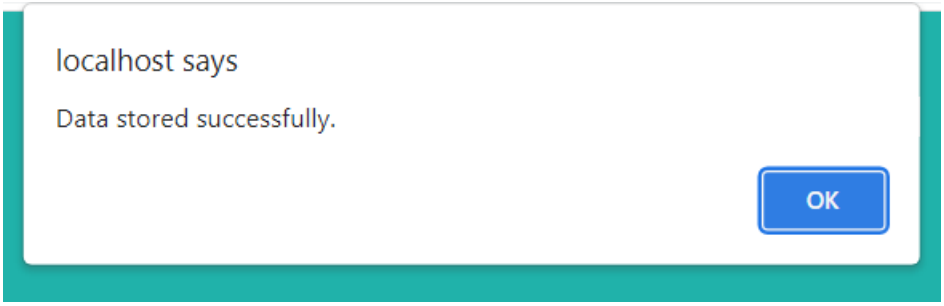


- This is the home page. As the first step suppose an employee logs in and enters his/her credentials say, **user1** and **user1** as username and password respectively. He will be redirected to the HTML form.

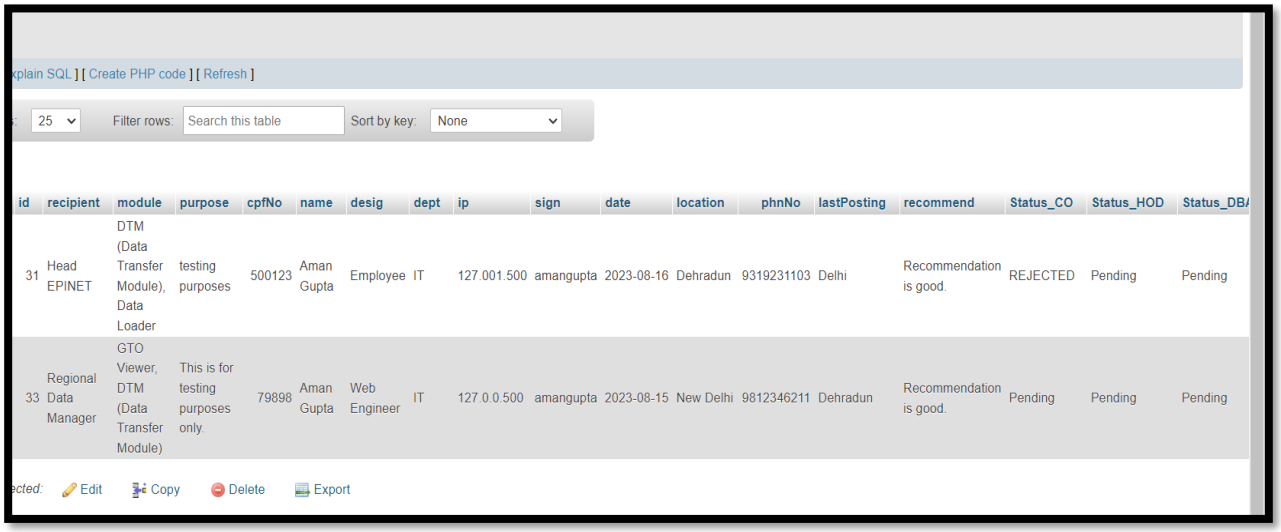
form1.html



- This is how the form looks like. When it is submitted, a pop up comes up showing success message. One can then log out of the window.



- Now the data has been inserted into the database.



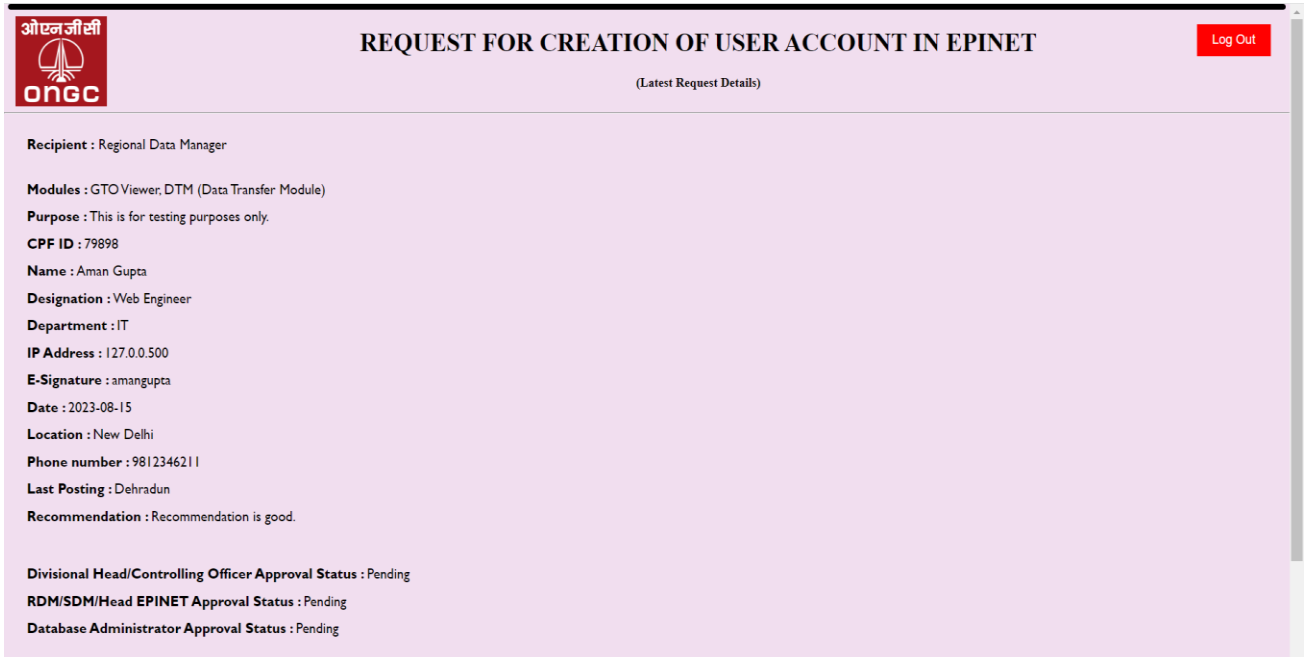
- Here, entry **number 33** is the latest entry in the database. And this latest form submission will be available to the Controlling officer, the Head EPINET and the DBA for review.

- One can see the default status for the 3 status fields is currently **'Pending'.**

Coming back to the login page, suppose the controlling officer wants to review the latest form submission. He/she will log in from their account credentials.

Example, he/she logs in with **ControllingOfficer** and **ControllingOfficer** as username and password respectively. He will be redirected to the form data display page.

Co.php

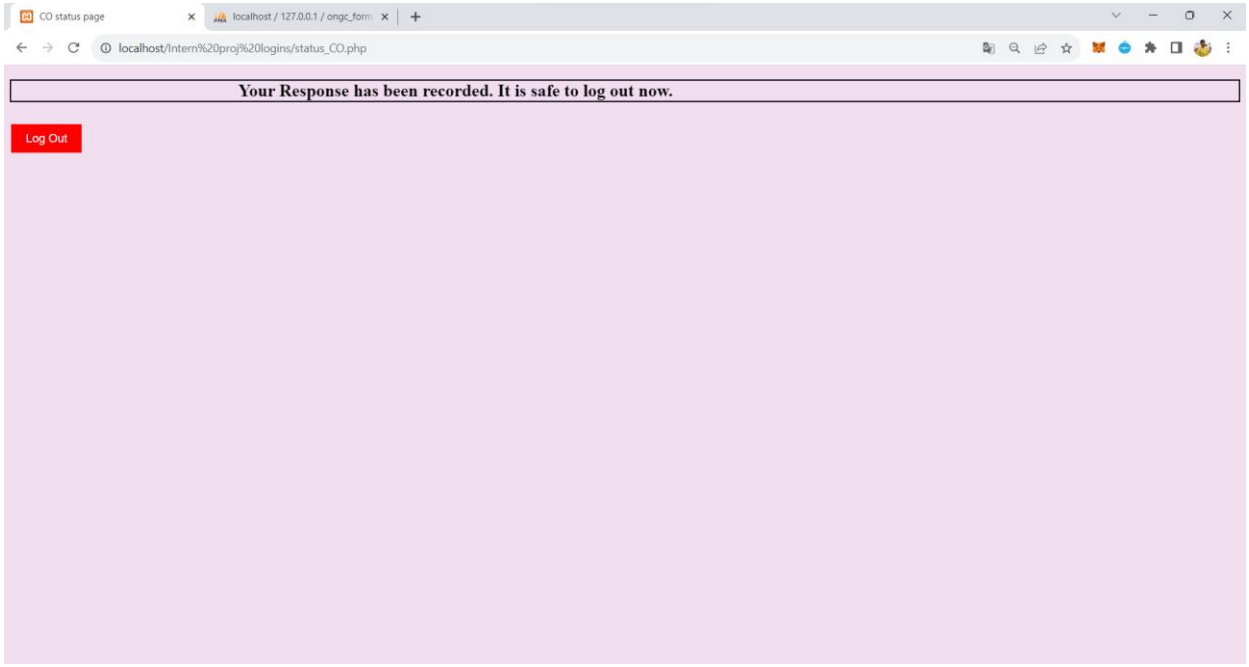


- When the controlling officer logs in he can see the latest form entry displayed on his page. It can be seen at the bottom of the page that 3 types of Approval status are displayed with all values as **'Pending'** and this is because no one has authorized the request yet.

- The controlling officer can scroll down to the page where he can find 3 options to validate the request at his level. These options are **'Approve', 'Reject' or 'Review'** in case he wants to refer to some third party and hold his approval status.

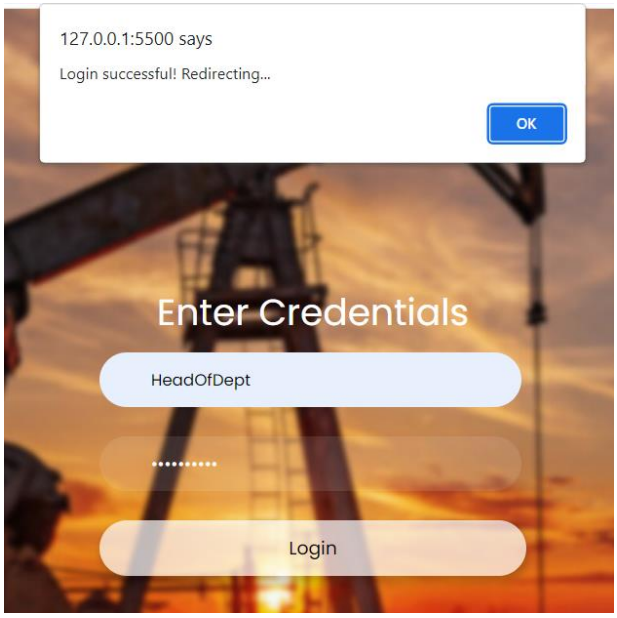
- As soon as the Controlling Officer updates his validation status and hits the Submit button, the status will be updated and whosoever will log in to see the data after that will see the updated status.

status_CO.php

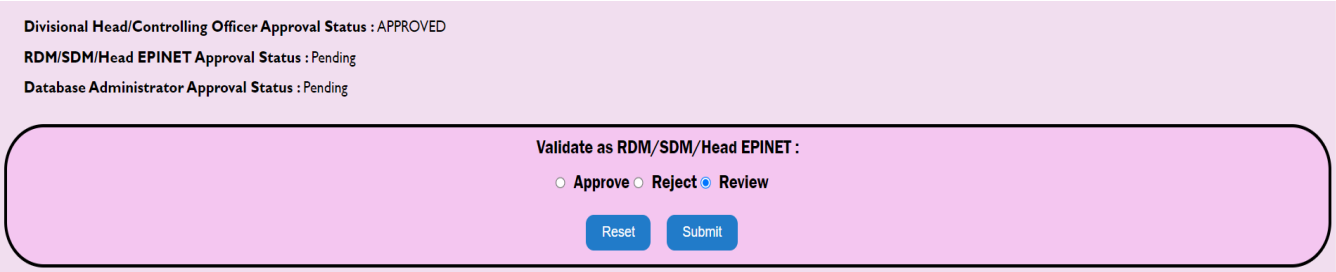


- Once the status is submitted by the Controlling Officer, A confirmation message will be displayed on the webpage as in the above screenshot.
- Further the user can log out of the window to the home login page.

- Now suppose the Head EPINET wants to view the request and validate at his front. From the login page he will log in with his credentials and will be directed to the display page, **'hod.php'** which is like the controlling officer but this time the updated status of the form will be displayed to him.



- The Head EPINET will be able to see the whole request form data as it was there in the controlling officer's account and the validation status for all three concerned authorities.



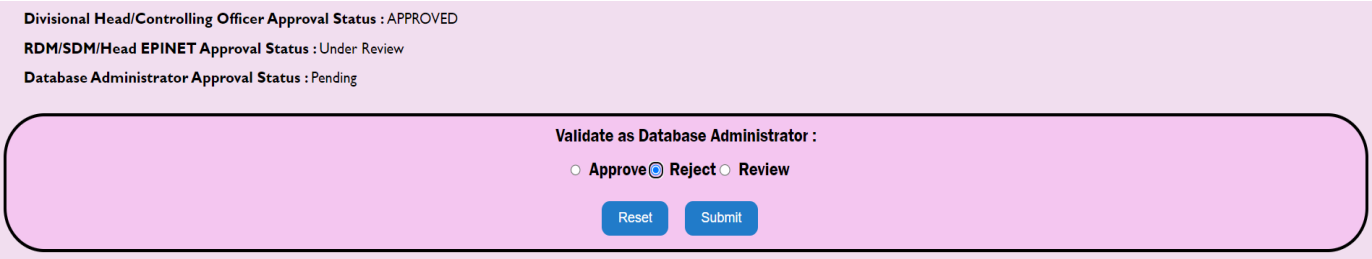
- Once he hits the submit button, the control will be directed to the status page **'status_hod.php'** where the SQL update query will get executed to update the status column called **Status_HOD** of the Head EPINET, in the database.

- The confirmation message will be displayed on the status page with a log out button like the one which was on the Controlling Officer's status page.

status_hod.php / status_dba.php

- Like the above 2 authentication processes for the controlling officer and for the Head EPINET, same procedure will happen once the **Database Administrator** logs in to review the form data. He will be able to see the status and update his validation as well. The update query will run through '**status_dba.php**' script.

dba.php



- The status for the 3 authorization levels is updated in the table **'request_ongc'** and can be seen as follows.

id	recipient	module	purpose	cpfNo	name	desig	dept	ip	sign	date	location	phnNo	lastPosting	recommend	Status_CO	Status_HOD	Status_DBA
31	Head EPINET	DTM (Data Transfer Module), Data Loader	testing purposes	500123	Aman Gupta	Employee	IT	127.001.500	amangupta	2023-08-16	Dehradun	9319231103	Delhi	Recommendation is good.	REJECTED	Pending	Pending
33	Regional Data Manager	GTO Viewer, DTM (Data Transfer Module)	This is for testing purposes only.	79898	Aman Gupta	Web Engineer	IT	127.0.0.500	amangupta	2023-08-15	New Delhi	9812346211	Dehradun	Recommendation is good.	APPROVED	Under Review	Pending

- The form request ID 33 is the latest form entry and as we can see the authorization status of the three senior authorities controlling officer, Head EPINET and the DBA can be seen in the last 3 columns of row ID 33.
- Columns named **Status_CO, Status_HOD** and the **Status_DBA** for the three respectively.

CONCLUSION

In conclusion, the internship project has been a transformative experience that has provided valuable insights, hands-on learning, and professional growth. Throughout the project, the objectives were met with dedication and a systematic approach, yielding significant findings and outcomes.

The project aimed to create an automated version of a process that is implemented manually in the current scenario. The main aim was to work on the very critical aspect of web development and exposure to the technology to implement a system which may potentially be deployed by the organization with some more specific improvisations.

Undertaking this project was not without challenges. Some of the challenges faced were mostly with respect to setting up the backend environment like many times the server was getting unresponsive, there were issues with maintaining a proper control flow between the web pages, selection, and implementation of the accurate data type for each of the form field values. But with determination and a broad-minded approach, these challenges were overcome. These experiences taught me the importance of adaptability and the value of a “not giving up” approach in achieving project goals.

One of the most rewarding aspects of this internship has been the continuous learning and personal development. The practical skills gained, such as self-motivation, an extroverted approach towards asking questions are assets that will undoubtedly contribute to my future endeavours. Moreover, the exposure to real-world scenarios and communicating directly with industry experts for the first time has honed my ability to think critically and communicate effectively.

The summer internship has helped me to get acquainted with most of the on-going activities of Corporate EPINET Data Centre at Dehradun and various responsibilities of this group. Involvement in development activities have also given me the opportunity to explore, learn and know various technologies, languages, and programming platforms. I am grateful to all those who contributed toward making my training a rich experience and to the staff of ONGC for being there whenever I needed their help.

BIBLIOGRAPHY

- ✓ Open AI Chat GPT 3.5
- ✓ www.geeksforgeeks.com
- ✓ www.w3schools.com
- ✓ https://www.w3schools.com/php/
- ✓ https://www.w3schools.com/js/
- ✓ https://stackoverflow.com/questions
- ✓ https://www.youtube.com/

❖ **GitHub repository Link for my Project source code:**
https://github.com/RuudraAmola/internship-ongc-July23