**LOCAL GOVERNMENT DIRECTORY MANAGEMENT SYSTEM**

Project report Submitted to

**Thanthai Periyar Government Arts and Science College (Autonomous), Tiruchirappalli–620023**

(Affiliated to Bharathidasan University, Tiruchirappalli–24)

in partial fulfillment of the requirements for the award of the degree of

**MASTER OF SCIENCE IN COMPUTER SCIENCE**

**By**

**V. SIVAKUMAR**

**(Reg.No:22PCS15)**

Under the guidance of

**Dr. R. PARIMALA M.Sc., D.C.P.A., M.Phil., Ph.D.,**

Associate Professor of Computer Science



**PG & RESEARCH DEPARTMENT OF COMPUTER SCIENCE**

**Thanthai Periyar Government Arts and Science College**

**(Autonomous), Tiruchirappalli–620023**

**APRIL –2024**

**PG & RESEARCH DEPARTMENT OF COMPUTER SCIENCE**

**Thanthai Periyar Government Arts and Science College (Autonomous),**

**Tiruchirappalli-23**



**CERTIFICATE**

This is to certify that the project work entitled “**LOCAL GOVERNMENT DIRECTORY MANAGEMENT SYSTEM”**, submitted in partial fulfillment of the requirements for the award of the Degree of **Master of Science in Computer Science** to Thanthai Periyar Government Arts and Science College(Autonomous), Tiruchirapalli-23 affiliated to Bharathidasan University, Tiruchirapalli-24 is a bonafide record of the work done by **V. SIVAKUMAR (Reg.No:22PCS15)** during the even semester of the academic year 2023-2024

Head of the Department Guide

Viva voce examination for this project work held on..........................

Internal Examiner External Examiner

## DECLARATION

**I,** **V. SIVAKUMAR (Reg.No:22PCS15)**, declared that the project entitled “**LOCAL GOVERNMENT DIRECTORY MANAGEMENT SYSTEM”** submitted to the P.G. & Research Department of Computer Science, Thanthai Periyar Government Arts & Science College (Autonomous), Tiruchirappalli -23 affiliated to Bharathidasan University, Tiruchirappalli-24 in partial fulfillment of the requirements for the award of the degree of **Master of Science in Computer Science** is the original project work done by me and it has not been submitted anywhere else for any other degree/diploma.

**V.SIVAKUMAR**

**(Reg.No:22PCS15)**

**ACKNOWLEDGEMENT**

I would like to express my sincere thanks to our honorable principal **Dr. P. S. VIJAYALAKSHMI, M.Com., M.B.A., M.Phil., Ph.D.,** Thanthai Periyar Government Arts and Science College (Autonomous), Tiruchirappalli who grant me permission to do my project and providing facilities to carry out this work.

I wish to express my sincere thanks to Mr. **T. KANNADASAN, M.C.A.,** Head of the Department of Computer Science, Thanthai Periyar Government Arts & Science College (Autonomous), Tiruchirappalli for his valuable support and helping me in various way throughout this project duration.

I wish to express my grateful thanks to my project guide   
**Dr. R. PARIMALA M.Sc., D.C.P.A., M.Phil., Ph.D.,** Associate Professor, Department of Computer Science, Thanthai Periyar Government Arts and Science College (Autonomous), Tiruchirappalli for her continuous supportive guidance and has been a great source of inspiration in guiding me throughout the project work.

I express my invaluable thanks to all the staff members of the department of Computer Science, Thanthai Periyar Government Arts & Science College (Autonomous), Tiruchirappalli, for their cooperation and encouragement that helped me a lot to bring this project in a successful way.

Finally, my heartfelt thanks are to my friends and parents who are the pillars and effort stone for each and every activity of mine. I thank my friends who supported and encouraged till the completion of the project work.

**ABSTRACT**

**“LOCAL GOVERNMENT DIRECTORY MANAGEMENT SYSTEM”** is a concept of an integral **digital** directory management system. The concept aims at an efficient and impeccable automated directory management system over network. The concept came to building up as an entity as it felt like good for many in the digital society the application replaces the conventional directory and personal profile management system. The digital application is reduced as much as possible to avoid errors while managing the data and ensure the security and protection data. This application is potential to be application would have a very wide range of application. Plugin accommodated into different websites as an add on extension. This system can lead to error free, secure, reliable and directory management system. This project is developed using PHP with MYSQL. The project has been divided into various modules, which performs different functions and there are inter-related with each other.

|  |  |  |
| --- | --- | --- |
| **CHAPTER** | **CONTENT** | **PAGE**  **NO.** |
| 1 | **INTRODUCTION** | 1 |
| 2 | **SYSTEM ANALYSIS**  2.1. Existing System  2.2. Disadvantages  2.3. Proposed System  2.4. Advantages | 2 |
| 3 | **SYSTEM CONFIGURATION**  3. 1. Hardware Requirement  3.2. Software Requirement  3.3 Software Description  3.3.1. Feasibility study | 5 |
| 4 | **SYSTEM DESCRIPTION**  4.1. Project description  4.2. Modules  4.2.1. Admin module  4.2.2. User module | 9 |
| 5 | **SYSTEM DESIGN**  5.1Data flow diagrams  5.1.1. System flow chart  5.2. Input design  5.3. Database design | 11 |
| 6 | **IMPLEMENTATION** | 17 |
| 7 | **SYSTEM TESTING** | 19 |
| 8 | **SOURCE CODE**  8.1 Output design  8.2. Screenshot | 64 |
|  | **CONCLUSION** | 73 |
|  | **FUTURE ENHANCEMENTS** | 74 |
|  | **BIBILOGRAPHY** | 75 |

**TABLE OF CONTENTS**

**CHAPTER-1**

**INTRODUCTION**

The conventional method of managing assembly directory information typically relies on paperwork and manual efforts. This process involves significant energy expenditure and relies heavily on the tenure of workers within an organization. However, this approach is fraught with challenges. Firstly, there's a high risk of records being misplaced or lost altogether. Secondly, the physical records are susceptible to decay over time, potentially leading to the loss of valuable information. Additionally, maintaining the privacy and security of these records presents a significant challenge. To address these issues, an automated application can be developed to streamline assembly directory management. By replacing traditional practices with software solutions, many of the inherent drawbacks of manual systems can be overcome. the adoption of an automated, web-oriented application for managing assembly directory information offers the fastest and most secure solution. By leveraging technology to streamline processes and enhance security, local governments can ensure efficient and effective management of vital information critical to their operations.

**CHATPTER- 2**

**SYSTEM ANALYSIS**

Process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is problem solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose. It is the process of studying a procedure or business to identify its goal and purposes and create systems and procedures that will efficiently achieve them. The field of system analysis relates closely to requirements analysis or to operations research. It is also "an explicit formal inquiry carried out to help a decision maker identify a better course of action and make a better decision than she might otherwise have made.

**2.1. Existing System**

In the existing system, the management of district data within the local government directory relies heavily on manual processes. Administrators are tasked with manually entering and managing this data, which may be stored in physical files or spreadsheets. Access to directory information is limited, and there is often no user-friendly interface for users to view or interact with the data. All tasks related to data insertion, updating, and deletion are handled solely by administrators, leading to potential bottlenecks and delays. Additionally, the existing system lacks robust data validation and error checking mechanisms, increasing the likelihood of inaccuracies or inconsistencies in the directory. Moreover, the manual handling of sensitive information poses security risks, as there is limited control over access and permissions. Overall, the existing system suffers from inefficiencies, limited accessibility, and security vulnerabilities, highlighting the need for a more streamlined and automated solution.

**2.2. DISADVANTAGES**

* + - Initial setup and development costs associated with building the web-based application and database.
    - Dependency on internet connectivity for users to access the system.
    - Potential resistance to change from users accustomed to manual processes.
    - Ongoing maintenance and updates required to ensure system functionality and security.
    - Risk of technical issues such as system downtime or data corruption.
    - Possibility of data privacy concerns if security measures are not implemented effectively.

**2.3. PROPOSED SYSTEM**

* Web-based application with a user-friendly interface for administrators and users.
* Centralized database for storing district data, including details on districts, parliamentary constituencies, villages, and blocks.
* Separate modules for administrators and users with appropriate access controls.
* Features for administrators to insert, update, and delete district data.
* Users can access and view all directory details, including districts, parliamentary, constituencies, villages, and blocks.
* Implement data validation and error checking mechanisms to ensure data accuracy and integrity.
* Implement security measures such as user authentication, access control, and encryption to protect sensitive information.

**2.4. ADVANTAGES**

* Improved efficiency and accuracy through automation of data management tasks.
* Enhanced accessibility for users to access directory information from anywhere with an internet connection.
* Centralized storage of data reduces the risk of data loss or duplication.
* Enhanced security measures protect sensitive information from unauthorized access or breaches.
* User-friendly interface improves usability and reduces training requirements.
* Scalable architecture allows for future expansion and integration with other systems.

## CHATPTER- 3

## SYSTEM CONFIGURATION

## SPECIFICATION REQUIREMENTS

Hardware and software specification gives the complete details about the hardware and software that were used to develop the system. Selection of hardware and software is very important factor for efficiency. Hardware specification specifies the requirements of the hardware to run the system. Software specification specifies the software requirements to develop the system in an efficient manner

## 3.1. HARDWARE REQUIREMENTS

Processor : AMD PRO A4-3350B APU with Radeon R4 Graphics 2.00 GHz

RAM : 4GB and above

Hard Disk :232GB

**3.2. SOFTWARE REQUIREMENT**

Operating System : Windows OS

Front End : HTML , CSS

Back End : PHP , MYSQL

Server : Xampp 7.1 and above

**3.3. SOFTWARE DESCRIPTION**

This whole project is created by the main two backend languages namely PHP and MySQL. The frontend used are HTML and CSS. The backend language provides database facilities for the front end. The back end stores the whole project data to the database.

**3.3.1.HTML**

HTML (Hypertext Markup Language) provides the structural framework for creating web pages within the application. It defines the content hierarchy, elements, and attributes that compose the user interface. HTML markup facilitates the organization and presentation of directory information, such as lists of districts, tables of constituency data, and forms for data entry. HTML works in conjunction with PHP to generate dynamic content based on user interactions, ensuring a seamless browsing experience for administrators and users. HTML facilitates the creation of structured and accessible web pages, ensuring compatibility across different browsers and devices.

**3.3.2 PHP**

PHP (Hypertext Preprocessor) serves as the backbone of the system, facilitating server-side scripting to generate dynamic web content. It enables the application to interact with the MySQL database, handle user requests, process form submissions, and execute business logic. PHP ensures that the directory management system is responsive and capable of delivering personalized experiences to users based on their interactions. PHP enables the creation of dynamic web pages with interactive features, enhancing user engagement and productivity. PHP applications may be susceptible to security vulnerabilities if not properly configured and maintained, requiring diligent attention to security best practices.

**3.3.3. MySQL**

MySQL (Relational Database Management System) serves as the database management system for storing and organizing directory data. It provides a robust and scalable platform for managing large volumes of structured data efficiently. MySQL's relational database model allows for the creation of tables to store information about districts, parliament constituencies, villages, blocks, and related entities. Through SQL queries, the application can retrieve, insert, update, and delete data seamlessly, ensuring data integrity and consistency. MySQL provides a reliable and scalable database solution for storing and managing directory data efficiently.

**3.3.4. CSS**

CSS (Cascading Style Sheets) is responsible for styling and designing the user interface of the application. It enables developers to define the visual presentation of HTML elements, including fonts, colors, layouts, and responsiveness. CSS ensures a consistent and visually appealing user experience across different devices and screen sizes. By separating the content from its presentation, CSS allows for easy customization and maintenance of the application's appearance, enhancing usability and accessibility. CSS allows for flexible and customizable styling of the application's user interface, ensuring a visually appealing and consistent design. CSS and HTML may require additional effort to ensure consistent rendering and compatibility across different web browsers and platforms, particularly in older versions.

**3.3.5. FEASIBILITY STUDY**

The system is developed using widely available technologies like PHP, MySQL, CSS, and HTML, making it technically feasible. The modular approach ensures scalability and maintainability, allowing for future enhancements and updates. The system fulfills all the identified objectives, indicating successful implementation of technical requirements. The system offers benefits such as secured data, faster processes, error-free operations, and better management capabilities. Evaluate the technical resources and capabilities required for developing the system using PHP, MySQL, CSS, and HTML. The system features a user-friendly interface, enabling administrators and users to perform operations without inconvenience. Minimal training is required for users to effectively utilize the system, contributing to its operational feasibility. Strategies are in place to manage organizational changes and ensure smooth adoption of the new system. The mission is to facilitate easy management and administration with capabilities to do insert, delete, update and view operation on constitutional details such as assembly in district, constituency, parliament, block panchayat and village panchayat. The system meets legal and regulatory requirements, safeguarding data integrity and confidentiality. Assess the projected timeline for developing and deploying the system, considering factors such as resource availability, development complexity, and potential delays. Implement robust security measures to protect against unauthorized access, data breaches, and cyber threats.

**Chapter-4**

**SYSTEM DESCRIPTION**

The Local Government Directory Management System is a robust web-based application designed to facilitate the efficient management of directory information for local government entities. Developed using PHP, MySQL, CSS, and HTML technologies, this system provides administrators with powerful tools for uploading, updating, and managing data related to districts, parliament constituencies, villages, and blocks. Additionally, users can access detailed information about these entities, ensuring transparency and accessibility.

**4.1. Project description**

The Local Government Directory Management System aims to streamline the management of directory information for local government entities, replacing manual processes with automated workflows. The system provides administrators with powerful tools for uploading, updating, and managing directory data, while also offering users convenient access to detailed information about districts, parliament constituencies, villages, and blocks.

**4.2. Modules**

* Admin module
* User module

**4.2.1. Admin module**

The Admin Module empowers administrators with comprehensive control over the system's functionalities, including data management, user management, reporting, and settings configuration. Admins can efficiently upload, update, and delete directory data, manage user accounts and permissions, generate reports on system activities, and customize system settings to meet specific requirements.

**4.2.2. User module**

The User Module provides users with convenient access to directory information, allowing them to search for specific entities, view detailed profiles, and provide feedback on the system's functionality. Users can authenticate themselves to access the system securely, access support resources to resolve any issues or queries they may encounter, and personalize their profiles according to their preferences.

**CHAPTER -5**

**SYSTEM DESIGN**

**5.1. DATA FLOW DIAGRAM**

A system flow chart is a physical design tool that shows in general terms the operations that will be performed on information in an information system. The arrows on a system flowchart show the direction that data will flow in around the system rather than the order in which the operations will be carried out.

index

Admin Login

User Login

Login

All Details

District wise

Constituency wise

Insert

Parliament wise

Update

Block wise

Delete

Village wise

**Admin Page**

Admin

Login

Admin Page

Insert

Update

Delete

**User Login**

User Login

All details

Block

Block

Village

Block

Parliament

District

Block wise

Village wise

Parliament

Constituency

District

Constituency

District

Village

District

Constituency

Village

Parliament

District

Block

Village

Parliament

Constituency

Block

Village

Constituency

Parliament

Parliament

wise

Constituency

wise

District wise

**SYSTEM DESIGN**

Systems design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development. The purpose of the System Design process is to provide sufficient detailed data and information about the system and its system elements to enable the implementation consistent with architectural entities as defined in models and views of the system architecture. We need to clarify the goal of the system. System design is such a vast topic; if we don't narrow it down to a specific goal, it will become complicated to design the system, especially for newbies. Sometimes constraints are good for the system. It helps to focus on the main feature you are trying to design. It clarifies ambiguities about the features of the system

**5.2 INPUT DESIGN**

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required. controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy in this project all the input data are validated and if any data violates any conditions. the user is warmed by a message. If the data satisfied all the conditions, it is transferred to the appropriate tables in the database. The user details are to be entered at the time of registration. A page is designed for this purpose which is user friendly and easy to use. The design is done such that user get appropriate messages when exceptions occur.

**Admin**: **Login**

Username and password are stored in a dictionary. When a user attempts to log in, the system checks if the entered username exists in the dictionary and then verifies if the associated password matches the input. If both credentials match, access is granted, allowing the user to perform admin tasks. This basic system lacks advanced features like encryption, error handling, and secure password management, which are essential for real-world applications to ensure robust security and usability.

**5.3. DATABASE DESIGN**

Database Design is a collection of processes that facilitate the designing

development, implementation and maintenance of enterprise data management systems. Properly designed database easy to maintain, improves data consistency and are cost effective in terms of disk storage space. The database designer decides how the data elements correlate and what data must be stored the main objectives of database design in DBMS are to produce logical and physical designs models of the proposed database system The logical model concentrates on the data requirements and the data to be stored independent of physical considerations. It does not concern itself with how the data will be stored or where it will be stored physical the physical data design model involves translating the logical DB design of the database onto physical media using hardware resources and software systems such as database management systems (DBMS). The new admin registration details will be stored on the user table in the database. In the user table contains six fields. They are ID, username, email, password, address and phone number. Fields we give the data type. VARCHAR is a variable length string data type, so it holds only the characters you assign to it. VARCHAR takes up I byte per character. +2 bytes to hold length information. INT stands for the integer that is a whole number. An integer can be written without a fractional component

**5.3.1. ASSEMBLY**

* **Table name: Assembly**
* **Primary key: D id**
* **Description:** This table stores details for the districts.

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELDS** | **DATA TYPE** | **SIZE** | **CONSTRINTS** |
| D id | INT | 5 | NOT NULL |
| District | VARCHAR | 25 | NOT NULL |
| Constitiuency | VARCHAR | 25 | NOT NULL |
| Parliament | VARCHAR | 25 | NOT NULL |
| Block | VARCHAR | 25 | NOT NULL |
| Village | VARCHAR | 25 | NOT NULL |

**5.3.2. LOGIN**

* **Table name: Login**
* **Primary key: Password**
* **Description:** This table stores the details for admin login username and password.

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELDS** | **DATA TYPE** | **LENGTH** | **ALLOW NULL** |
| Username | VARCHAR | 25 | NOT NULL |
| Password | VARCHAR | 10 | NOT NULL |

**CHAPTER -6**

**IMPLEMENTATION**

Gather and analyze requirements for the system, including functionalities and user roles. Design the system architecture, database schema, and user interface based on the requirements. Develop backend functionalities using PHP and MySQL for data management and frontend interfaces using HTML and CSS. Deploy the system to a production environment, ensuring proper configurations and monitoring. Provide training for users and administrators and offer ongoing support to address any issues or questions.

**6.1. MODULES**

* Admin module
* User module

**6.1.1. Admin module**

* Admins can insert, update, and delete directory data, such as districts, parliament constituencies, villages, and blocks.
* Access control mechanisms ensure that only authorized admins can perform administrative tasks.
* Admins can monitor system activities, generate reports, and configure system settings.

**6.1.2. User module**

* Users can view directory details, including districts, parliament constituencies, villages, and blocks.
* The user interface provides functionalities for searching, filtering, and sorting directory information.
* Users can access reports and analytics generated by the system and provide feedback on the user experience.

**CHAPTER -7**

**SYSTEM TESTING**

Software testing is a critical element of software quantity assurance and represents the ultimate review of specialization, design and coding the objective of the system testing is to ensure that all the individual programs are working and the programs link together to meet the requirements specified and to ensure that the computer system and the associated clerical and other procedures work together.

Testing is vital to the success of the system. System testing marks a logical assumption that all the parts of the system are correct; the goal will be successfully achieved. Special test data is input for processing and the result are examined to locate unexpected results.

* Testing is the process of executing a program with the indent of finding an error.
* A good test case is one of that has a higher probability of finding a yet undiscovered error.
* A successful test is one that uncovers a yet undiscovered error.

System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences.

Testing is vital to the success of the system. System testing makes a logical assumption that if all parts of the system are correct, the goal will be successfully achieved.

Testing is actually a series of different test whose primary purpose is to fully exercise the computer-based system. Testing begins at the module level and works toward the integration of the entire computer-based system.

**TYPE OF TESTING**

Four Testing Strategies that are often adopted by the software.

* Unit Testing
* Integration Testing
* Validation Testing
* System Testing

This system was tested using Unit Testing and Integration Testing Strategies to test the project because there were the most relevant approaches for this project.

**Unit Testing**

In the unit testing the analyst tests the program making up a system. The software units in a system are the modules and routines that are assembled and integrated to perform a specific function. In a large system, many modules on different levels are needed.

Unit testing can be performed from the bottom up starting with the smallest and lowest level modules and proceeding one at a time. For each module in a bottom-up testing, a short program executes the module and provides the needed data.

The entire system is divided into several modules and is developed individually. Hence, unit testing is conducted to individual modules.

**Integration Testing**

After the unit test, each module is gradually integrated to form one final system. Hence, the objective of integration testing is to take unit tested modules and build a final program structure. In this project, modules are combined to find the overall performance of the system.

Integration testing is a systematic technique for constructing the program structure while conducting test to uncover errors associate with interfacing. Objectives are used to take unit test modules and built program structure that has been directed by design.

**Black Box Testing**

This method treats the coded module as a black box. The module runs with inputs that are likely to cause errors. Then the output is checked to see if any error occurred. This method cannot be used to test all errors, because some errors may depend on the code or algorithm used to implement the module.

**White Box Testing**

* White box testing, sometimes called glass-box testing is a test case design method that uses the control structure of the procedural design to derive test cases. Using white box testing methods, the software engineer can derive test cases that
  + Guarantee that all independent paths with in a module have been exercised at least once.
  + Exercise all logical decisions on their true and false sides.
* Executive all loops at their boundaries and within their operational bounds and
  + Exercise internal data structure to assure their validity.

**Validation Testing**

There are two types of testing Alpha testing and Beta Testing. Alpha testing it is conducted at the developer site by end user. Controlled by developer. Conducted controlled environment. It is conducted at user’s site. Conducted time application software environment.

**Acceptance Testing**

The objective of the acceptance test is to tell the user about the validity and reliability of the system, it verifies whether the system operates as specified and the integrity of important data is maintained. User motivation is very important is very important for the successful performance of the system.

All the modules were tested individually using both test data and live data. After each module was ascertained that it was working correctly and it had been “integrated” with the system. Again the system was tested as a whole/ we hold the system tested with different types of users. The system design, data flow diagrams, procedures etc, were well documented so that the system can be easily maintained and upgraded by any computer professional at a late.

**CHAPTER -8**

**SOURCE CODE**

**Index.php**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Assembly Data</title>

<style>

body {

margin: 0;

padding: 0;

font-family: Arial, sans-serif;

background-color: #f0f0f0;

}

.container {

display: flex;

justify-content: center;

align-items: center;

min-height: 100vh;

}

.content {

background-color: #ffffff;

padding: 40px;

border-radius: 10px;

box-shadow: 0px 0px 20px rgba(0, 0, 0, 0.1);

text-align: center;

}

h2 {

color: #333;

margin-bottom: 20px;

}

h1 {

margin-bottom: 30px;

color: cyan;

}

span {

color: #333;

font-weight: bold;

}

. btn {

display: inline-block;

background-color: #007bff;

color: #fff;

padding: 12px 24px;

text-decoration: none;

border-radius: 5px;

margin: 10px;

transition: background-color 0.3s ease;

border: none;

cursor: pointer;

}

.btn:hover {

background-color: #0056b3;

}

.btn-admin {

background-color: #dc3545;

}

.btn-user {

background-color: #28a745;

}

. btn-group {

margin-top: 30px;

}

.btn-group a {

display: block;

margin-bottom: 10px;

}

.btn-group a:last-child {

margin-bottom: 0;

}

</style>

</head>

<body>

<div class="container">

<div class="content">

<h2>Local Government Directory</h2>

<h1>Welcome, <span>User</span>!</h1>

<div class="btn-group">

<a href="login.html" class="btn btn-admin">Admin Login</a>

<a href="user\_login.html" class="btn btn-user">User Login</a>

</div>

</div>

</div>

</body>

</html>

**Login.php**

<?php

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

if (isset($\_POST["user"]) && isset($\_POST["pass"])) {

$username = $\_POST["user"];

$password = $\_POST["pass"];

if ($username === "admin" && $password === "password") {

// Redirect to a welcome page or perform further actions

header ("Location: admin\_page.html");

exit ();

} else {

// If credentials are invalid, redirect back to the login page with an error message

header ("Location: admin\_page.html");

exit ();

}

}

}

header ("Location: login.html");

exit ();

?>

**Welcome.php**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

</head>

<body>

<h1>Login Successful</h1>

</body>

</html>

**Admin.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Login</title>

<link rel="stylesheet" type="text/css" href="style.css">

<style>

body {

margin: 0;

padding: 0;

font-family: Arial, sans-serif;

background-color: #f0f0f007;

}

.container {

display: flex;

justify-content: center;

align-items: center;

min-height: 100vh;

}

.content {

background-color: #ffffff17;

padding: 40px;

border-radius: 10px;

box-shadow: 0px 0px 20px rgba (0, 0, 0, 0.1);

text-align: center;

}

h2 {

color: #333;

margin-bottom: 20px;

}

h1 {

margin-bottom: 30px;

color: cyan;

}

span {

color: #333;

font-weight: bold;

}

. btn {

display: inline-block;

background-color: black;

color: #fff;

padding: 12px 24px;

text-decoration: none;

border-radius: 5px;

margin: 10px;

transition: background-color 0.3s ease;

}

.btn:hover {

background-color: cyan;

}

.btn-group {

margin-top: 30px;

}

. btn-group a {

display: block;

margin-bottom: 10px;

}

.btn-group a:last-child {

margin-bottom: 0;

}

</style>

</head>

<body>

<div class="container">

<div class="content">

<h2>Local Government Directory</h2>

<h1>Welcome, <span>User</span>! </h1>

<div class="btn-group">

<a href="insert\_data.html" class="btn">INSERT</a>

</div>

<div class="btn-group">

<a href="update\_data.html" class="btn">UPDATE</a>

</div>

<div class="btn-group">

<a href="delete\_data.html" class="btn">DELETE</a>

</div>

</div>

</div>

</body>

</html>

**Insert data.php**

<?php

$servername = "localhost";

$username = "root";

$password = "";

$database = "database1";

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

if ($conn->connect\_error) {

die ("Connection failed: “. $conn->connect\_error);

}

if ($\_SERVER["REQUEST\_METHOD"] == "POST" && isset($\_POST['d\_id']) && isset($\_POST['dist']) && isset($\_POST['const']) && isset($\_POST['vpt']) && isset($\_POST['bpt']) && isset($\_POST['cpc'])) {

$d\_id = $\_POST['d\_id'];

$dist = $\_POST['dist'];

$const = $\_POST['const'];

$vpt = $\_POST['vpt'];

$bpt = $\_POST['bpt'];

$cpc = $\_POST['cpc'];

$sql = "INSERT INTO assembly (d\_id, dist, const, vpt, bpt, cpc) VALUES ('$d\_id', '$dist', '$const', '$vpt', '$bpt', '$cpc')";

if ($conn->query($sql) === TRUE) {

echo "Data inserted successfully";

} else {

echo "Error: " . $sql. "<br> “. $conn->error;

}

} else {

echo "Error: All form fields are required.";

}

$conn->close();

?>

**Insert data.html**

<!DOCTYPE html>

<html>

<head>

<title>Login</title>

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<style>

body {

font-family: Arial, sans-serif;

background-color: #f7f7f7;

margin: 0;

padding: 0;

}

#form {

width: 80%;

background-color: #fff;

border-radius: 8px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

max-width: 400px; /\* Limiting maximum width for larger screens \*/

margin: 50px auto;

padding: 20px;

}

h1 {

text-align: center;

color: #333;

margin-bottom: 20px; /\* Adding some space below the heading \*/

}

label {

display: block;

margin-bottom: 10px;

color: #555;

}

input[type="text"], input[type="password"] {

width: calc(100% - 20px);

padding: 10px;

margin-bottom: 15px; /\* Reducing margin bottom for input fields \*/

border: 1px solid #ccc;

border-radius: 4px;

}

input[type="submit"] {

width: 100%;

padding: 10px;

background-color: #4caf50;

color: #fff;

border: none;

border-radius: 4px;

cursor: pointer;

transition: background-color 0.3s ease;

}

input[type="submit"]:hover {

background-color: #6c45a0;

}

</style>

</head>

<body>

<div id="form">

<h1>Login Form</h1>

<form name="form" action="data\_insert.php" method="POST">

<input type="text" name="d\_id" placeholder="ID"><br>

<input type="text" name="dist" placeholder="District" required><br>

<input type="text" name="const" placeholder="Assembly constituney name" required><br>

<input type="text" name="vpt" placeholder="Village panchayat" required><br>

<input type="text" name="bpt" placeholder="Block panchayat" required><br>

<input type="text" name="cpc" placeholder="Parliment constituency" required><br>

<input type="submit" value="Submit">

</form>

</div>

</body>

</html>

**Update data.php**

<?php

$servername = "localhost";

$username = "root";

$password = "";

$database = "database1";

// Create connection

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

// Check connection

if ($conn->connect\_error) {

die ("Connection failed: ". $conn->connect\_error);

}

// Check if the form is submitted

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

// Get form data

$d\_id = $\_POST['d\_id'];

$dist = $\_POST['dist'];

$const = $\_POST['const'];

$vpt = $\_POST['vpt'];

$bpt = $\_POST['bpt'];

$cpc = $\_POST['cpc'];

// SQL query to update data in the 'assembly' table

$sql = "UPDATE assembly SET dist='$dist', const='$const', vpt='$vpt', bpt='$bpt', cpc='$cpc' WHERE d\_id='$d\_id'";

if ($conn->query($sql) === TRUE) {

echo "Data updated successfully";

} else {

echo "Error: “. $sql. "<br> “. $conn->error;

}

}

// Close the database connection

$conn->close ();

?>

**Update data.html**

<!DOCTYPE html>

<html>

<head>

<title>Data</title>

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" type="text/css" href="style.css">

<style>

body {

font-family: Arial, sans-serif;

background-color: #f7f7f7;

margin: 0;

padding: 0;

}

#form1 {

background-color: #fff;

border-radius: 8px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

width: 80%;

max-width: 400px; /\* Limiting maximum width for larger screens \*/

margin: 50px auto;

padding: 20px;

}

h1 {

text-align: center;

color: #333;

margin-bottom: 20px; /\* Adding some space below the heading \*/

}

label {

display: block;

margin-bottom: 10px;

color: #555;

}

input[type="text"] {

width: calc(100% - 20px);

padding: 10px;

margin-bottom: 15px; /\* Reducing margin bottom for input fields \*/

border: 1px solid #ccc;

border-radius: 4px;

}

input[type="submit"] {

width: 100%;

padding: 10px;

background-color: #4caf50;

color: #fff;

border: none;

border-radius: 4px;

cursor: pointer;

transition: background-color 0.3s ease;

}

input[type="submit"]:hover {

background-color: #45a049;

}

</style>

</head>

<body>

<div id="form1">

<h1>Assembly Data</h1>

<form name="form1" action="data\_update.php" method="POST" onsubmit="return isValid()">

<input type="text" name="d\_id" placeholder="ID"><br>

<input type="text" name="dist" placeholder="District" required><br>

<input type="text" name="const" placeholder="Assembly constituney name" required><br>

<input type="text" name="vpt" placeholder="Village panchayat" required><br>

<input type="text" name="bpt" placeholder="Block panchayat" required><br>

<input type="text" name="cpc" placeholder="Corresponding parliment constituency" required><br>

<input type="submit" id="btn" value="Update" name="submit">

</form>

</div>

</body>

</html>

**Delete data.php**

<?php

$servername = "localhost";

$username = "root";

$password = "";

$database = "database1";

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

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$d\_id = $\_POST['d\_id'];

$sql = "DELETE FROM assembly WHERE d\_id='$d\_id'";

if ($conn->query($sql) === TRUE) {

echo "Data deleted successfully";

} else {

echo "Error: " . $sql . "<br>" . $conn->error;

}

}

$conn->close();

?>

**Delete data.html**

<!DOCTYPE html>

<html>

<head>

<title>Data</title>

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" type="text/css" href="style.css">

<style>

body {

font-family: Arial, sans-serif;

background-color: #f7f7f7;

margin: 0;

padding: 0;

}

#form1 {

background-color: #fff;

border-radius: 8px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

width: 80%;

max-width: 400px; /\* Limiting maximum width for larger screens \*/

margin: 50px auto;

padding: 20px;

}

h1 {

text-align: center;

color: #333;

margin-bottom: 20px; /\* Adding some space below the heading \*/

}

label {

display: block;

margin-bottom: 10px;

color: #555;

}

input[type="text"] {

width: calc(100% - 20px);

padding: 10px;

margin-bottom: 15px; /\* Reducing margin bottom for input fields \*/

border: 1px solid #ccc;

border-radius: 4px;

}

input[type="submit"] {

width: 100%;

padding: 10px;

background-color: #ff5722;

color: #fff;

border: none;

border-radius: 4px;

cursor: pointer;

transition: background-color 0.3s ease;

}

input[type="submit"]:hover {

background-color: #f4511e;

}

</style>

</head>

<body>

<div id="form1">

<h1>Assembly Data</h1>

<form name="form1" action="data\_delete.php" method="POST" onsubmit="return isValid()">

Enter district ID

<input type="text" id="d\_id" name="d\_id" placeholder="ID" required>

<input type="submit" id="btn" value="Delete" name="submit">

</form>

</div>

</body>

</html

**Userlogin.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>User Login</title>

<style>

body {

margin: 0;

padding: 0;

font-family: Arial, sans-serif;

background-color: #f0f0f007;

}

.container {

display: flex;

justify-content: center;

align-items: center;

min-height: 100vh;

}

.content {

background-color: #ffffff17;

padding: 40px;

border-radius: 10px;

box-shadow: 0px 0px 20px rgba(0, 0, 0, 0.1);

text-align: center;

}

h2 {

color: #333;

margin-bottom: 20px;

}

h1 {

margin-bottom: 30px;

color: cyan;

}

span {

color: #333;

font-weight: bold;

}

. btn {

display: inline-block;

background-color: black;

color: #fff;

padding: 12px 24px;

text-decoration: none;

border-radius: 5px;

margin: 10px;

transition: background-color 0.3s ease;

}

.btn:hover {

background-color: cyan;

}

.btn-group {

margin-top: 30px;

}

.btn-group a {

display: block;

margin-bottom: 10px;

}

.btn-group a:last-child {

margin-bottom: 0;

}

</style>

</head>

<body>

<div class="container">

<div class="content">

<h2>Local Government Directory</h2>

<h1>Welcome, <span>User</span>! </h1>

<div class="btn-group">

<a href="select\_data.html" class="btn">All Details</a>

<a href="district\_data.html" class="btn">Districtwise</a>

<a href="constiuency\_data.html" class="btn">Constiuencywise</a>

<a href="village\_data.html" class="btn">Villagewise</a>

<a href="block\_data.html" class="btn">Blockwise</a>

<a href="p-data.html" class="btn">Parliamentwise</a>

</div>

</div>

</div>

</body>

</html>

**District data.php**

<?php

$servername = "localhost";

$username = "root";

$password = "";

$database = "database1";

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

if ($conn->connect\_error) {

die ("Connection failed: ". $conn->connect\_error);

}

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

// Get form data

// $d\_id = $\_POST['d\_id'];

// SQL query to delete data from the 'assembly' table

$sql = "SELECT \* FROM `assembly` ";

if ($res = mysqli\_query ($conn, $sql)) {

echo "<div style='margin: 0 auto; width: 80%;'>"; // Centering the table

echo "<table style='border-collapse: collapse; width: 100%;'>";

echo "<thead>";

echo "<tr>";

echo "<th style='padding: 12px 8px; background-color: #f2f2f2; color: #333; text-align: left;'>District ID</th>";

echo "<th style='padding: 12px 8px; background-color: #f2f2f2; color: #333; text-align: left;'>District</th>";

echo "<th style='padding: 12px 8px; background-color: #f2f2f2; color: #333; text-align: left;'>Constituency</th>";

echo "<th style='padding: 12px 8px; background-color: #f2f2f2; color: #333; text-align: left;'>Village Panchayat</th>";

echo "<th style='padding: 12px 8px; background-color: #f2f2f2; color: #333; text-align: left;'>Block Panchayat</th>";

echo "<th style='padding: 12px 8px; background-color: #f2f2f2; color: #333; text-align: left;'>Parliament Constituency</th>";

echo "</tr>";

echo "</thead>";

echo "<tbody>";

while ($row = mysqli\_fetch\_assoc($res)) {

echo "<tr>";

echo "<td style='padding: 10px 8px; border-bottom: 1px solid #ddd;'>".$row['d\_id']."</td>";

echo "<td style='padding: 10px 8px; border-bottom: 1px solid #ddd;'>".$row['dist']."</td>";

echo "<td style='padding: 10px 8px; border-bottom: 1px solid #ddd;'>".$row['const']."</td>";

echo "<td style='padding: 10px 8px; border-bottom: 1px solid #ddd;'>".$row['vpt']."</td>";

echo "<td style='padding: 10px 8px; border-bottom: 1px solid #ddd;'>".$row['bpt']."</td>";

echo "<td style='padding: 10px 8px; border-bottom: 1px solid #ddd;'>".$row['cpc']."</td>";

echo "</tr>";

}

echo "</tbody>";

echo "</table>";

echo "</div>";

} else {

echo "Error: ". $sql . "<br>" . $conn->error;

}

}

$conn->close ();

?>

**Constiuencywise.php**

<?php

$servername = "localhost";

$username = "root";

$password = "";

$database = "database1";

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

if ($conn->connect\_error) {

die ("Connection failed: ". $conn->connect\_error);

}

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

// Get the entered

$Constiuencywise = $\_POST['Constiuencywise'];

echo " entered Constiuencywise is $Constiuencywise";

// SQL query to retrieve data for the entered

$sql = "SELECT dist, const, vpt, bpt, cpc FROM `assembly` WHERE const='$Constiuencywise'";

// Execute the query

$result = mysqli\_query ($conn, $sql);

if ($result) {

echo "<div style='margin: 0 auto; width: 80%;'>"; // Inline CSS for centering

echo "<h2 style='text-align: center;'>Assembly Information for $Constiuencywise</h2>"; // Centering the heading

echo "<table style='border-collapse: collapse; width: 100%;'>";

echo "<thead>";

echo "<tr>";

echo "<th style='padding: 12px 8px; background-color: #f2f2f2; color: #333; text-align: left;'>District</th>";

echo "<th style='padding: 12px 8px; background-color: #f2f2f2; color: #333; text-align: left;'>Village Panchayat</th>";

echo "<th style='padding: 12px 8px; background-color: #f2f2f2; color: #333; text-align: left;'>Block Panchayat</th>";

echo "<th style='padding: 12px 8px; background-color: #f2f2f2; color: #333; text-align: left;'>Parliament Constituency</th>";

echo "</tr>";

echo "</thead>";

echo "<tbody>";

while ($row = mysqli\_fetch\_assoc($result)) {

echo "<tr>";

echo "<td style='padding: 10px 8px; border-bottom: 1px solid #ddd;'>".$row['dist']."</td>";

echo "<td style='padding: 10px 8px; border-bottom: 1px solid #ddd;'>".$row['vpt']."</td>";

echo "<td style='padding: 10px 8px; border-bottom: 1px solid #ddd;'>".$row['bpt']."</td>";

echo "<td style='padding: 10px 8px; border-bottom: 1px solid #ddd;'>".$row['cpc']."</td>";

echo "</tr>";

}

echo "</tbody>";

echo "</table>";

echo "</div>";

} else {

echo "Error: ". $sql. "<br>". $conn->error;

}

}

$conn->close ();

?>

**Villagewise.php**

<?php

$servername = "localhost";

$username = "root";

$password = "";

$database = "database1";

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

if ($conn->connect\_error) {

die ("Connection failed: ". $conn->connect\_error);

}

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

// Get the entered district

$village = $\_POST['village'];

echo " entered village is $village";

// SQL query to retrieve data for the entered district

$sql = "SELECT dist, const, bpt, cpc FROM `assembly` WHERE vpt='$village'";

// Execute the query

$result = mysqli\_query ($conn, $sql);

if ($result) {

echo "<div style='margin: 0 auto; width: 80%;'>"; // Inline CSS for centering

echo "<h2 style='text-align: center;'>Assembly Information for $village</h2>"; // Centering the heading

echo "<table style='border-collapse: collapse; width: 100%;'>";

echo "<thead>";

echo "<tr>";

echo "<th style='padding: 12px 8px; background-color: #f2f2f2; color: #333; text-align: left;'>District</th>";

echo "<th style='padding: 12px 8px; background-color: #f2f2f2; color: #333; text-align: left;'>Constituency</th>";

echo "<th style='padding: 12px 8px; background-color: #f2f2f2; color: #333; text-align: left;'>Block Panchayat</th>";

echo "<th style='padding: 12px 8px; background-color: #f2f2f2; color: #333; text-align: left;'>Parliament Constituency</th>";

echo "</tr>";

echo "</thead>";

echo "<tbody>";

while ($row = mysqli\_fetch\_assoc($result)) {

echo "<tr>";

echo "<td style='padding: 10px 8px; border-bottom: 1px solid #ddd;'>".$row['dist']."</td>";

echo "<td style='padding: 10px 8px; border-bottom: 1px solid #ddd;'>".$row['const']."</td>";

echo "<td style='padding: 10px 8px; border-bottom: 1px solid #ddd;'>".$row['bpt']."</td>";

echo "<td style='padding: 10px 8px; border-bottom: 1px solid #ddd;'>".$row['cpc']."</td>";

echo "</tr>";

}

echo "</tbody>";

echo "</table>";

echo "</div>";

} else {

echo "Error: ". $sql. "<br>". $conn->error;

}

}

// Close the database connection

$conn->close ();

?>

**Blockwise.php**

<?php

$servername = "localhost";

$username = "root";

$password = "";

$database = "database1";

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

if ($conn->connect\_error) {

die ("Connection failed: ". $conn->connect\_error);

}

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

// Get the entered district

$block = $\_POST['block'];

echo " entered block is $block";

// SQL query to retrieve data for the entered district

$sql = "SELECT dist, const, vpt, bpt, cpc FROM `assembly` WHERE bpt='$block'";

// Execute the query

$result = mysqli\_query ($conn, $sql);

if ($result) {

echo "<div style='margin: 0 auto; width: 80%;'>"; // Inline CSS for centering

echo "<h2 style='text-align: center;'>Assembly Information for $block</h2>";

echo "<table style='border-collapse: collapse; width: 100%;'>";

echo "<thead>";

echo "<tr>";

echo "<th style='padding: 12px 8px; background-color: #f2f2f2; color: #333; text-align: left;'>District</th>";

echo "<th style='padding: 12px 8px; background-color: #f2f2f2; color: #333; text-align: left;'>Constituency</th>";

echo "<th style='padding: 12px 8px; background-color: #f2f2f2; color: #333; text-align: left;'>Village Panchayat</th>";

echo "<th style='padding: 12px 8px; background-color: #f2f2f2; color: #333; text-align: left;'>Parliament Constituency</th>";

echo "</tr>";

echo "</thead>";

echo "<tbody>";

while ($row = mysqli\_fetch\_assoc($result)) {

echo "<tr>";

echo "<td style='padding: 10px 8px; border-bottom: 1px solid #ddd;'>".$row['dist']."</td>";

echo "<td style='padding: 10px 8px; border-bottom: 1px solid #ddd;'>".$row['const']."</td>";

echo "<td style='padding: 10px 8px; border-bottom: 1px solid #ddd;'>".$row['vpt']."</td>";

echo "<td style='padding: 10px 8px; border-bottom: 1px solid #ddd;'>".$row['cpc']."</td>";

echo "</tr>";

}

echo "</tbody>";

echo "</table>";

echo "</div>";

} else {

echo "Error: ". $sql. "<br>". $conn->error;

}

}

$conn->close ();

?>

**Parliamentwise.php**

<?php

$servername = "localhost";

$username = "root";

$password = "";

$database = "database1";

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

if ($conn->connect\_error) {

die ("Connection failed: ". $conn->connect\_error);

}

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

// Get the entered

$par = trim($\_POST['P']);

echo " Entered Parliament is $par";

// SQL query to retrieve data for the entered

$sql = "SELECT dist, const, vpt, bpt, cpc FROM `assembly` WHERE cpc='$par'";

$result = mysqli\_query ($conn, $sql);

if ($result) {

echo "<div style='margin: 0 auto; width: 80%;'>";

echo "<h2 style='text-align: center;'>Assembly Information for $par</h2>";

echo "<table style='border-collapse: collapse; width: 100%;'>";

echo "<thead>";

echo "<th style='padding: 12px 8px; background-color: #f2f2f2; color: #333; text-align: left;'>District</th>";

echo "<th style='padding: 12px 8px; background-color: #f2f2f2; color: #333; text-align: left;'>Village Panchayat</th>";

echo "<th style='padding: 12px 8px; background-color: #f2f2f2; color: #333; text-align: left;'>Block Panchayat</th>";

echo "<th style='padding: 12px 8px; background-color: #f2f2f2; color: #333; text-align: left;'> Constituency</th>";

echo "</tr>";

echo "</thead>";

echo "<tbody>";

while ($row = mysqli\_fetch\_assoc($result)) {

echo "<tr>";

echo "<td>".$row['dist']."</td>";

echo "<td style='padding: 10px 8px; border-bottom: 1px solid #ddd;'>".$row['vpt']."</td>";

echo "<td style='padding: 10px 8px; border-bottom: 1px solid #ddd;'>".$row['bpt']."</td>";

echo "<td style='padding: 10px 8px; border-bottom: 1px solid #ddd;'>".$row['const']."</td>";

echo "</tr>";

}

echo "</tbody>";

echo "</table>";

echo "</div>";

} else {

echo "Error: ". $sql . "<br>". $conn->error;

}

}

$conn->close ();

?>

**Parliamentwise.html**

<!DOCTYPE html>

<html>

<head>

<title>Data</title>

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<style>

body {

font-family: Arial, sans-serif;

background-color: #f7f7f7;

margin: 0;

padding: 0;

}

#div11 {

background-color: #ffffff;

border-radius: 8px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

width: 80%;

max-width: 400px; /\* Limiting maximum width for larger screens \*/

margin: 50px auto;

padding: 20px;

}

h1 {

text-align: center;

color: #333333;

margin-bottom: 20px; /\* Adding some space below the heading \*/

}

label {

display: block;

margin-bottom: 10px;

color: #555555;

font-size: 16px;

}

input[type="text"] {

width: calc(100% - 20px);

padding: 10px;

margin-bottom: 15px; /\* Reducing margin bottom for input fields \*/

border: 1px solid #cccccc;

border-radius: 4px;

font-size: 16px;

}

input[type="submit"] {

width: 100%;

padding: 10px;

background-color: #4caf50;

color: #ffffff;

border: none;

border-radius: 4px;

cursor: pointer;

transition: background-color 0.3s ease;

font-size: 16px;

}

input[type="submit"]:hover {

background-color: #45a049;

}

</style>

</head>

<body>

<div id="div11">

<h1>Assembly Data</h1>

<form name="form1" action="p\_wise.php" method="POST">

<input type="text" id="P" name="P" placeholder="P" required><br>

<input type="submit" id="btn" value="parliamentwise" name="submit">

</form>

</div>

</body>

</html>

**Districtwise.html**

<!DOCTYPE html>

<html>

<head>

<title>Data</title>

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<style>

body {

font-family: Arial, sans-serif;

background-color: #f7f7f7;

margin: 0;

padding: 0;

}

#div11 {

background-color: #ffffff;

border-radius: 8px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

width: 80%;

max-width: 400px; /\* Limiting maximum width for larger screens \*/

margin: 50px auto;

padding: 20px;

}

h1 {

text-align: center;

color: #333333;

margin-bottom: 20px; /\* Adding some space below the heading \*/

}

label {

display: block;

margin-bottom: 10px;

color: #555555;

}

input[type="text"] {

width: calc(100% - 20px);

padding: 10px;

margin-bottom: 15px; /\* Reducing margin bottom for input fields \*/

border: 1px solid #cccccc;

border-radius: 4px;

}

input[type="submit"] {

width: 100%;

padding: 10px;

background-color: #4caf50;

color: #ffffff;

border: none;

border-radius: 4px;

cursor: pointer;

transition: background-color 0.3s ease;

}

input[type="submit"]:hover {

background-color: #45a049;

}

</style>

</head>

<body>

<div id="div11">

<h1>Assembly Data</h1>

<form name="form1" action="data\_district\_wise.php" method="POST" >

<input type="text" id="district" name="district" placeholder="District" required><br>

<input type="submit" id="btn" value="Districtwise" name="submit">

</form>

</div>

</body>

</html>

**Constinuencywise.html**

<!DOCTYPE html>

<html>

<head>

<title>Data</title>

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<style>

body {

font-family: Arial, sans-serif;

background-color: #f7f7f7;

margin: 0;

padding: 0;

}

#div11 {

background-color: #ffffff;

border-radius: 8px;

box-shadow: 0 0 10px rgba (0, 0, 0, 0.1);

width: 80%;

max-width: 400px; /\* Limiting maximum width for larger screens \*/

margin: 50px auto;

padding: 20px;

}

h1 {

text-align: center;

color: #333333;

margin-bottom: 20px; /\* Adding some space below the heading \*/

}

label {

display: block;

margin-bottom: 10px;

color: #555555;

font-size: 16px;

}

input[type="text"] {

width: calc (100% - 20px);

padding: 10px;

margin-bottom: 15px; /\* Reducing margin bottom for input fields \*/

border: 1px solid #cccccc;

border-radius: 4px;

font-size: 16px;

}

input[type="submit"] {

width: 100%;

padding: 10px;

background-color: #4caf50;

color: #ffffff;

border: none;

border-radius: 4px;

cursor: pointer;

transition: background-color 0.3s ease;

font-size: 16px;

}

input[type="submit"]: hover {

background-color: #45a049;

}

</style>

</head>

<body>

<div id="div11">

<h1>Assembly Data</h1>

<form name="form1" action="data\_Constiuency\_wise.php" method="POST">

<input type="text" id="Constiuencywise" name="Constiuencywise" placeholder="Constiuency Name" required><br>

<input type="submit" id="btn" value="Constiuencywise" name="submit">

</form>

</div>

</body>

</html>

**Villagewise.html**

<!DOCTYPE html>

<html>

<head>

<title>Data</title>

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<style>

body {

font-family: Arial, sans-serif;

background-color: #f7f7f7;

margin: 0;

padding: 0;

}

#div11 {

background-color: #ffffff;

border-radius: 8px;

box-shadow: 0 0 10px rgba (0, 0, 0, 0.1);

width: 80%;

max-width: 400px; /\* Limiting maximum width for larger screens \*/

margin: 50px auto;

padding: 20px;

}

h1 {

text-align: center;

color: #333333;

margin-bottom: 20px; /\* Adding some space below the heading \*/

}

label {

display: block;

margin-bottom: 10px;

color: #555555;

font-size: 16px;

}

input[type="text"] {

width: calc (100% - 20px);

padding: 10px;

margin-bottom: 15px; /\* Reducing margin bottom for input fields \*/

border: 1px solid #cccccc;

border-radius: 4px;

font-size: 16px;

}

input[type="submit"] {

width: 100%;

padding: 10px;

background-color: #4caf50;

color: #ffffff;

border: none;

border-radius: 4px;

cursor: pointer;

transition: background-color 0.3s ease;

font-size: 16px;

}

input[type="submit"]: hover {

background-color: #45a049;

}

</style>

</head>

<body>

<div id="div11">

<h1>Assembly Data</h1>

<form name="form1" action="data\_village\_wise.php" method="POST">

<input type="text" id="village" name="village" placeholder="Village Name" required><br>

<input type="submit" id="btn" value="villagewise" name="submit">

</form>

</div>

</body>

</html>

**Blockwise.html**

<!DOCTYPE html>

<html>

<head>

<title>Data</title>

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<style>

body {

font-family: Arial, sans-serif;

background-color: #f7f7f7;

margin: 0;

padding: 0;

}

#div11 {

background-color: #fff;

border-radius: 8px;

box-shadow: 0 0 10px rgba (0, 0, 0, 0.1);

width: 80%;

max-width: 400px; /\* Limiting maximum width for larger screens \*/

margin: 50px auto;

padding: 20px;

}

h1 {

text-align: center;

color: #333;

margin-bottom: 20px; /\* Adding some space below the heading \*/

}

label {

display: block;

margin-bottom: 10px;

color: #555;

}

input[type="text"] {

width: calc (100% - 20px);

padding: 10px;

margin-bottom: 15px; /\* Reducing margin bottom for input fields \*/

border: 1px solid #ccc;

border-radius: 4px;

}

input[type="submit"] {

width: 100%;

padding: 10px;

background-color: #4caf50;

color: #fff;

border: none;

border-radius: 4px;

cursor: pointer;

transition: background-color 0.3s ease;

}

input[type="submit"]: hover {

background-color: #45a049;

}

</style>

</head>

<body>

<div id="div11">

<h1>Assembly Data</h1>

<form name="form1" action="data\_block\_wise.php" method="POST" >

<input type="text" id="block" name="block" placeholder="Block" required><br>

<input type="submit" id="btn" value="Blockwise" name="submit">

</form>

</div>

</body>

</html>

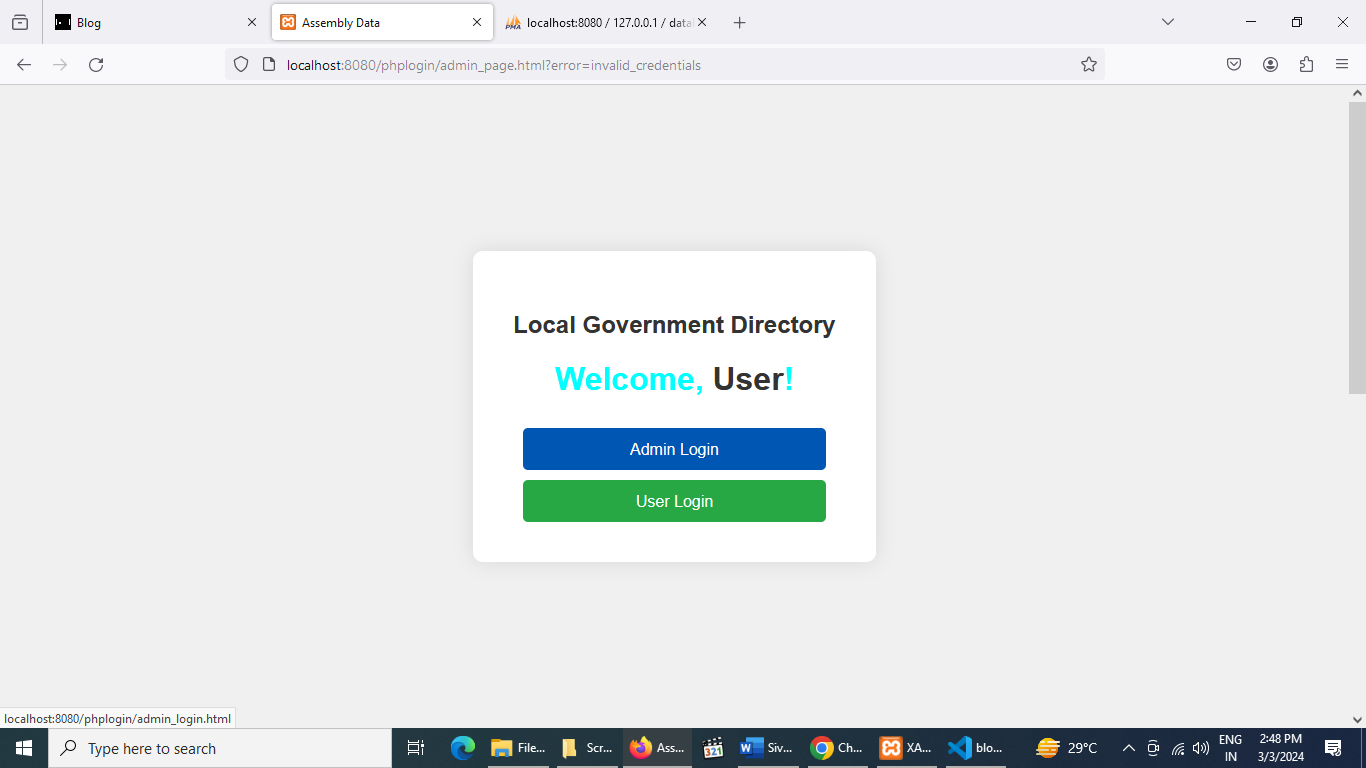
**8.1. OUTPUT DESIGN**

Design output is a drawing or specification or manufacturing instruction. design outputs describe all the components, parts, and pieces that go into your medical device. Design outputs describe all assemblies and sub-assemblies of your product.

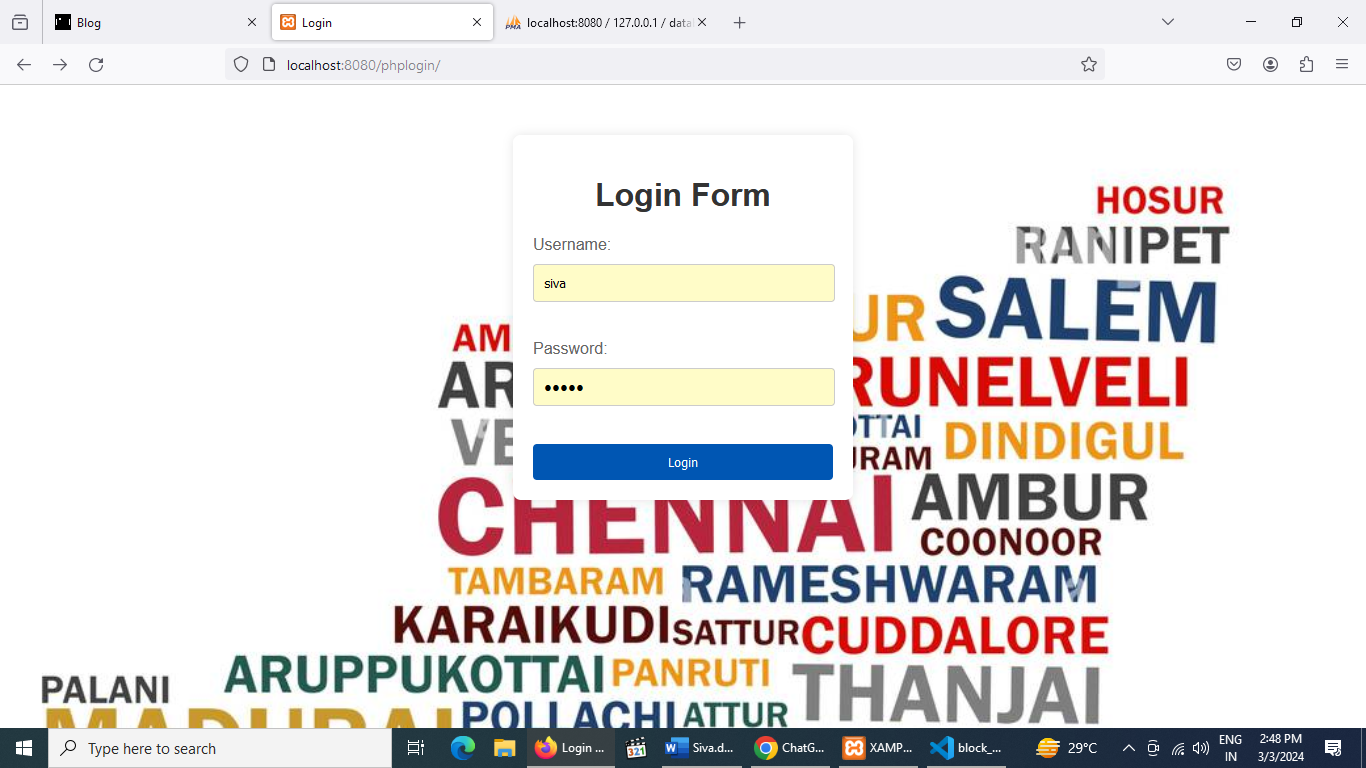
Objectives of output design the objectives design are designing output to fit the user delivering the appropriate quantity of output making sure the output is where it is needed providing the output on time choosing the right output method designing output to serve the intended purpose. Designing computer output should proceed in an organized, well thought our manner, the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively.

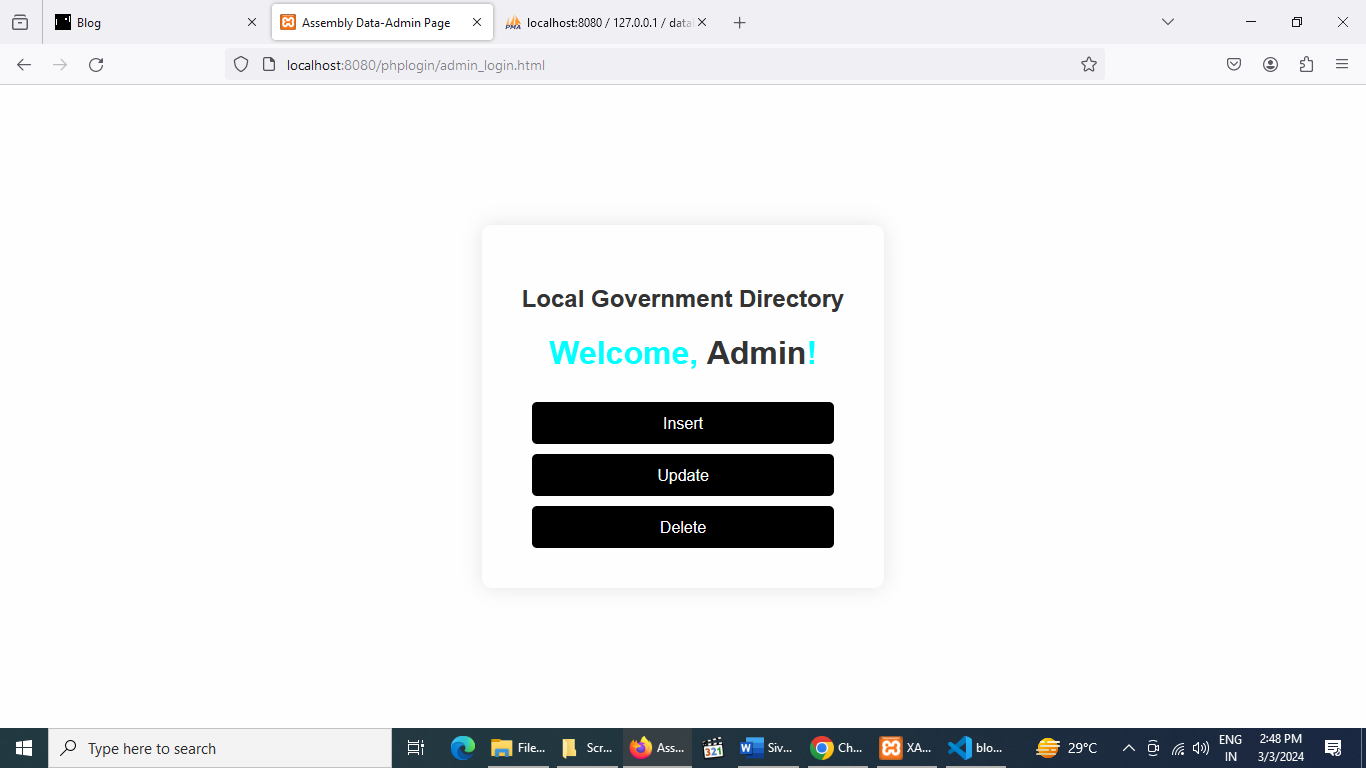
**8.2. SCREENSHOT**

**INDEX**

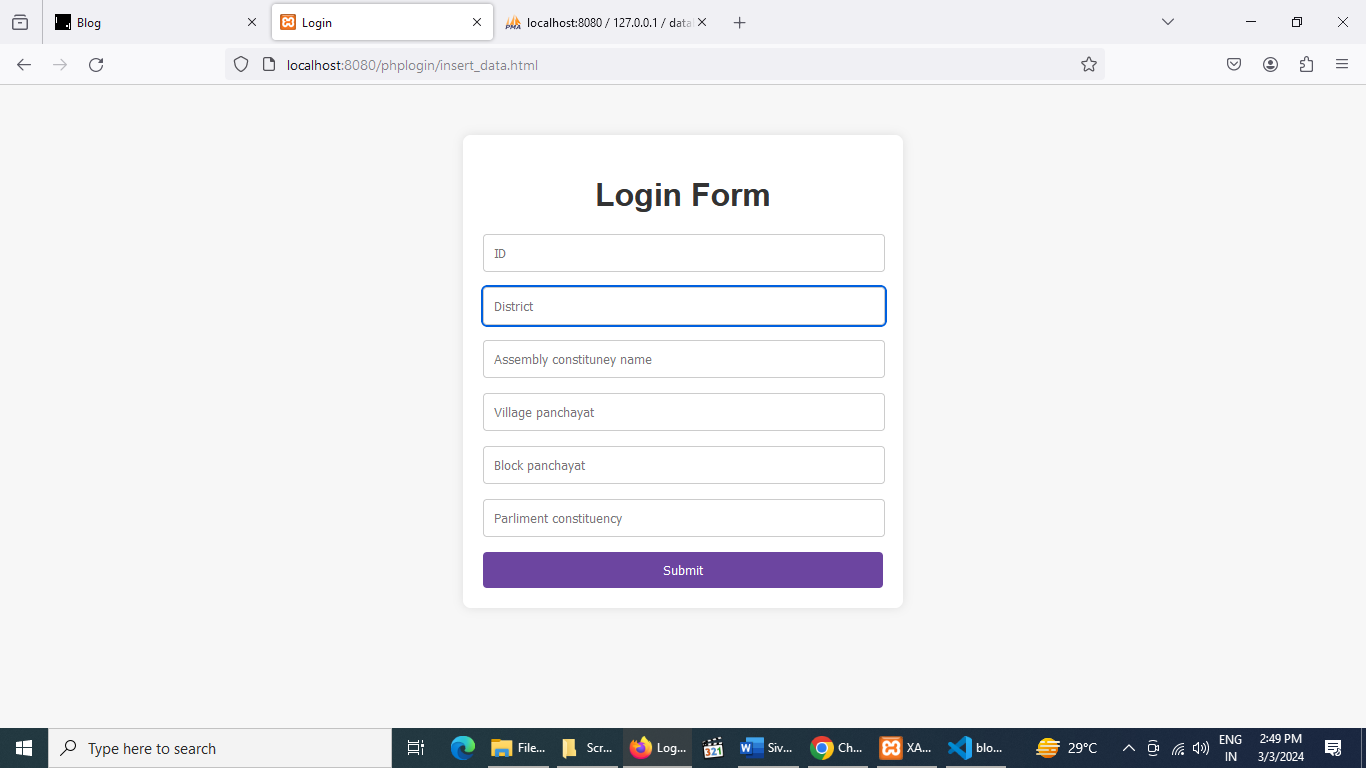


**ADMIN LOGIN**

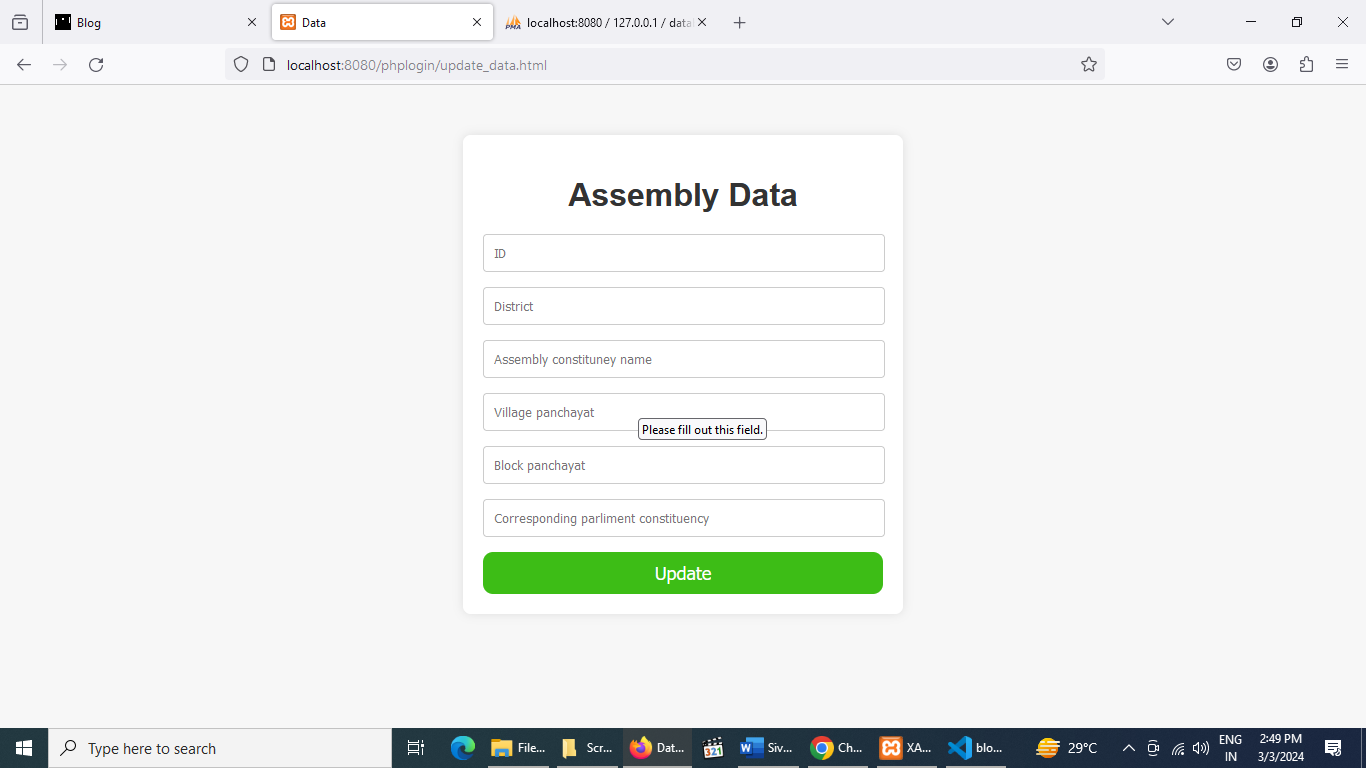
**Admin login page**



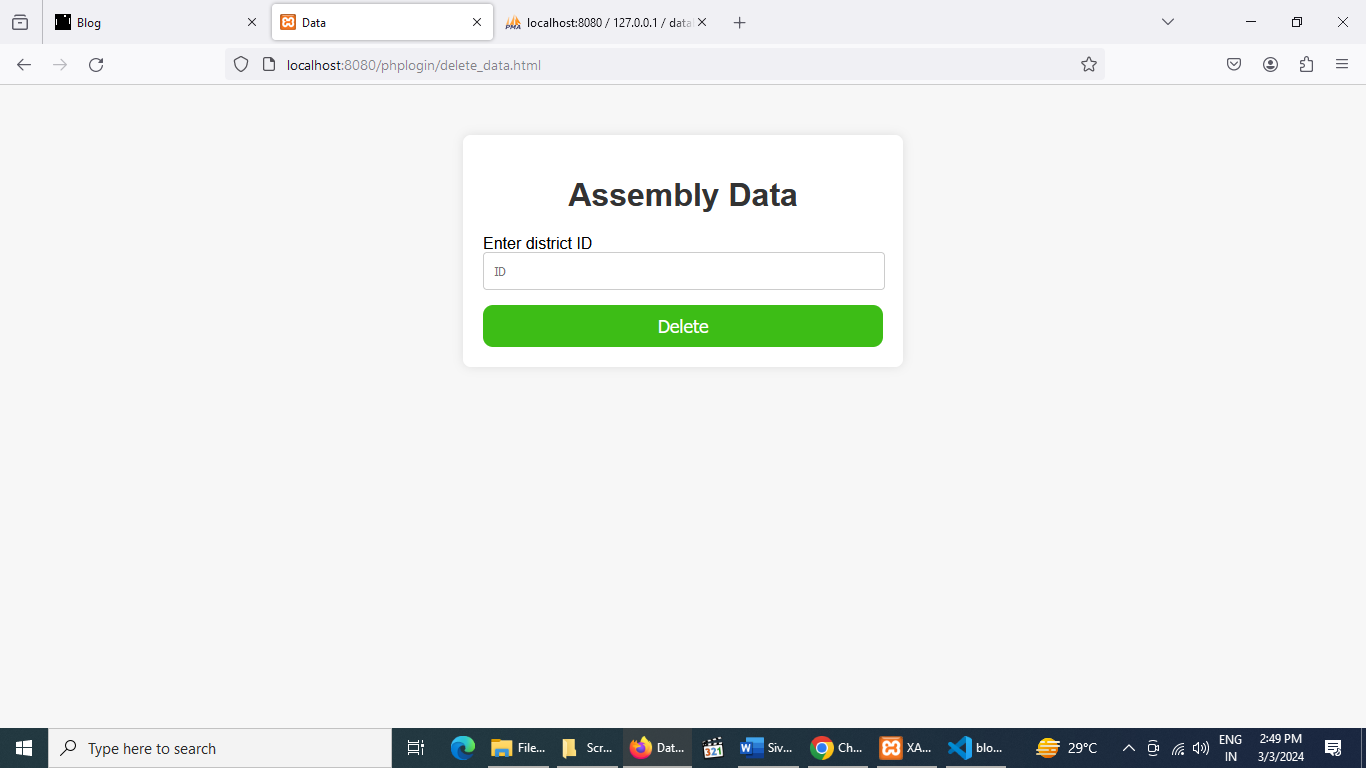
**Insert page**



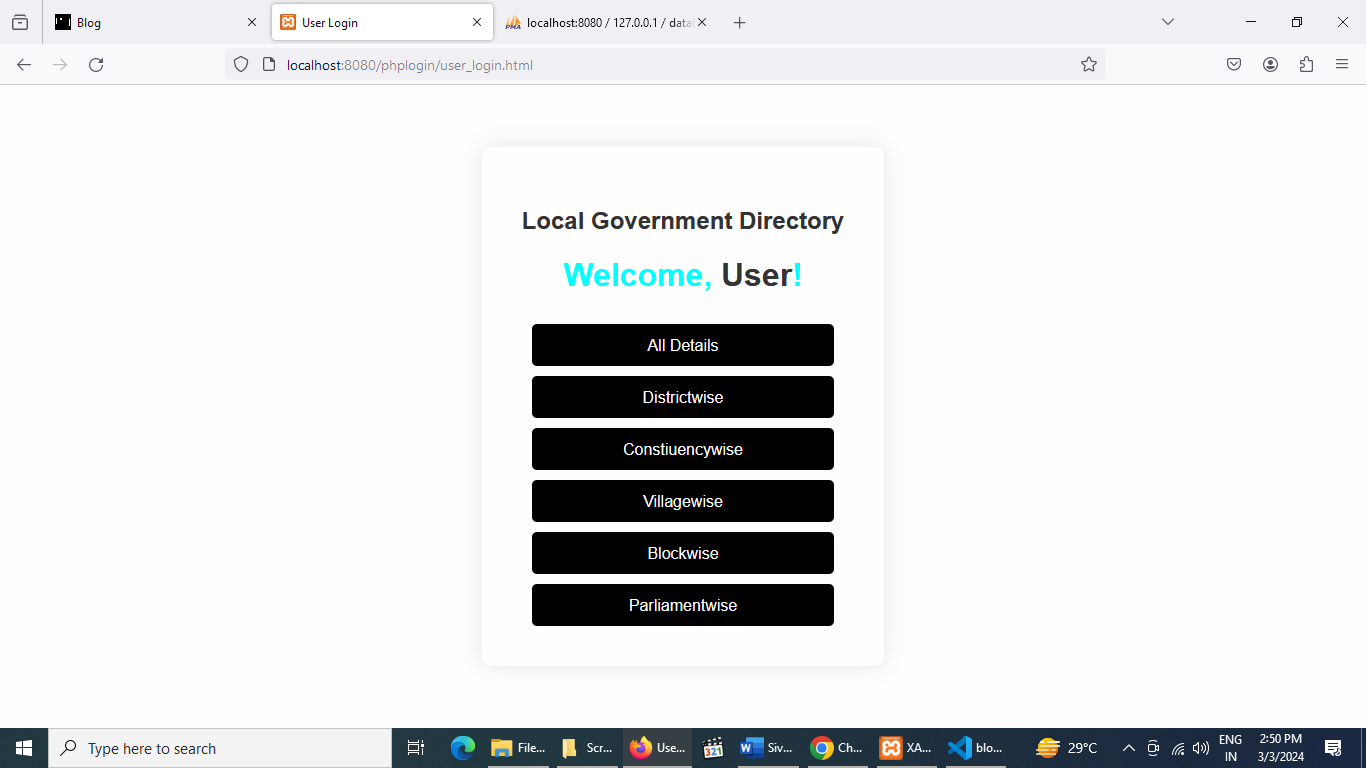
**Update page**

****

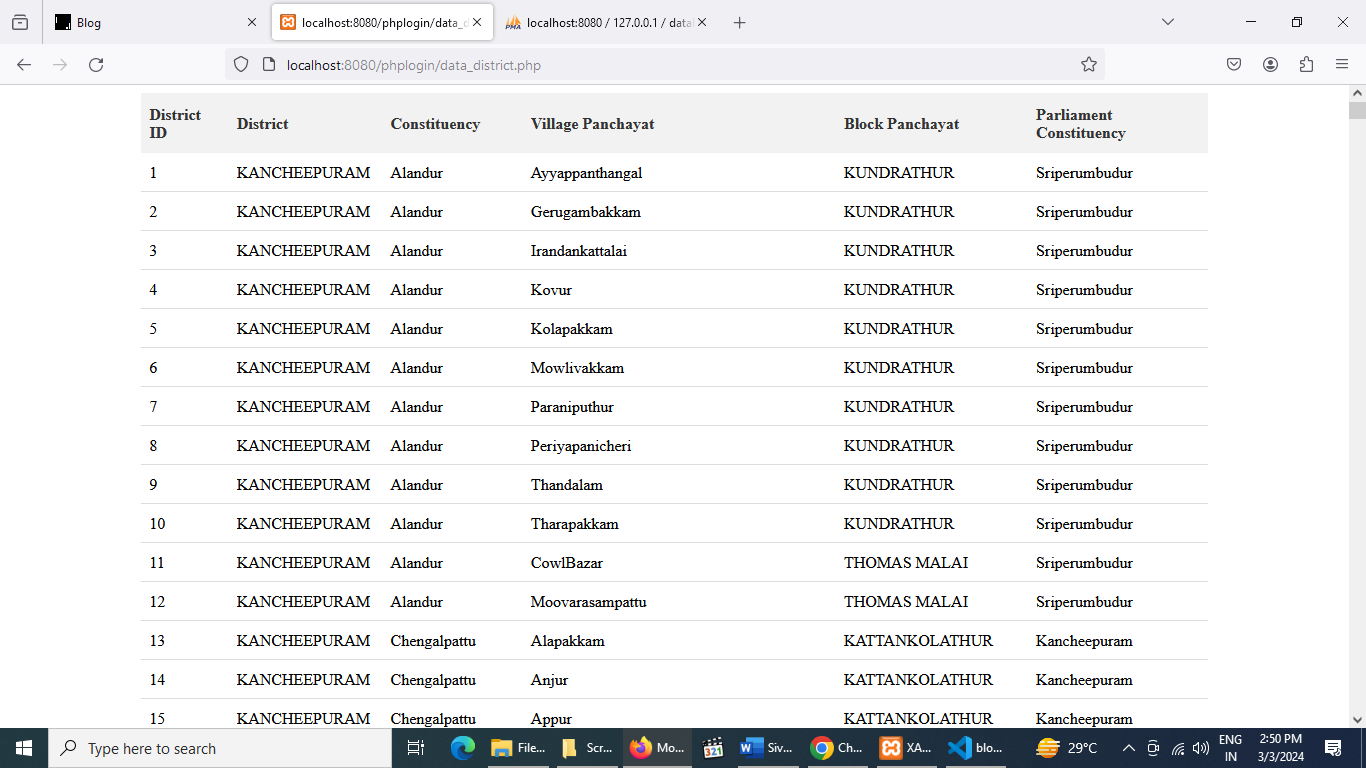
**Delete page**

****

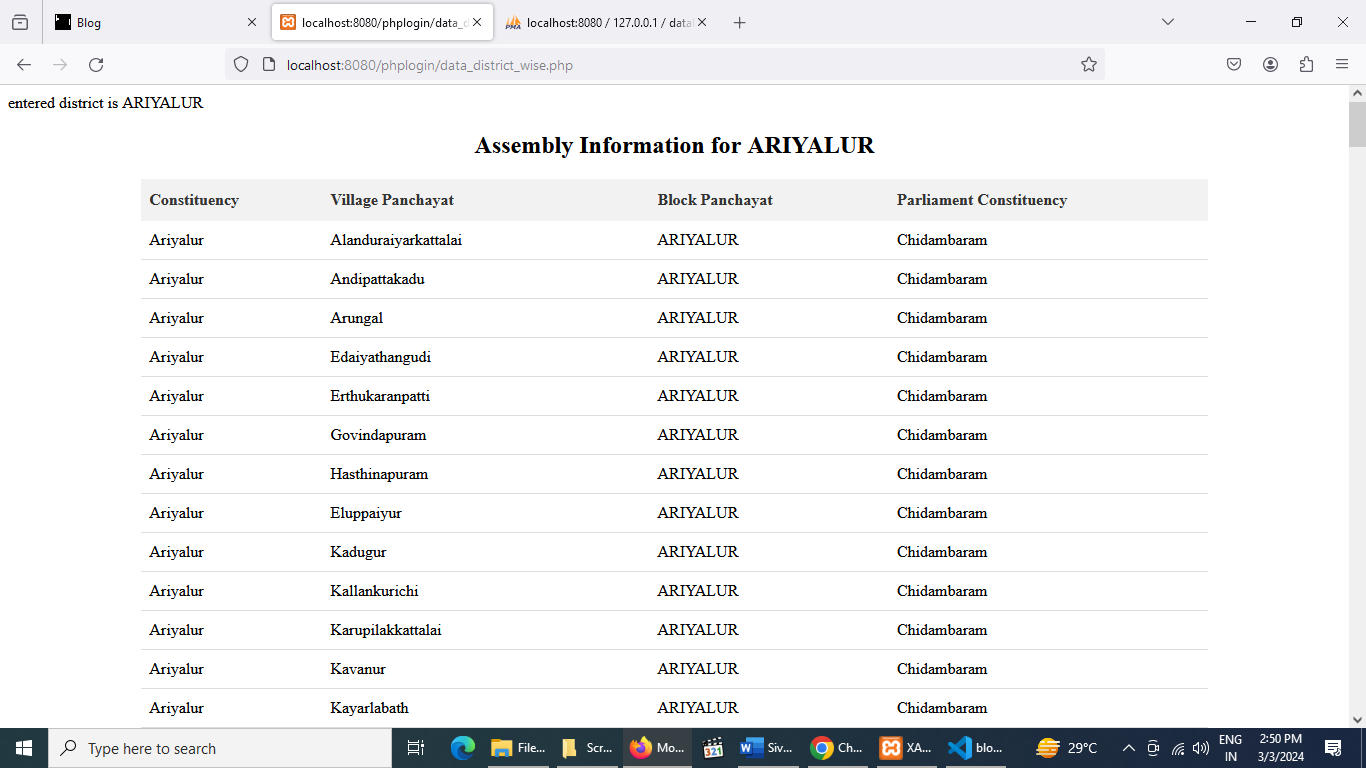
**User login page**

****

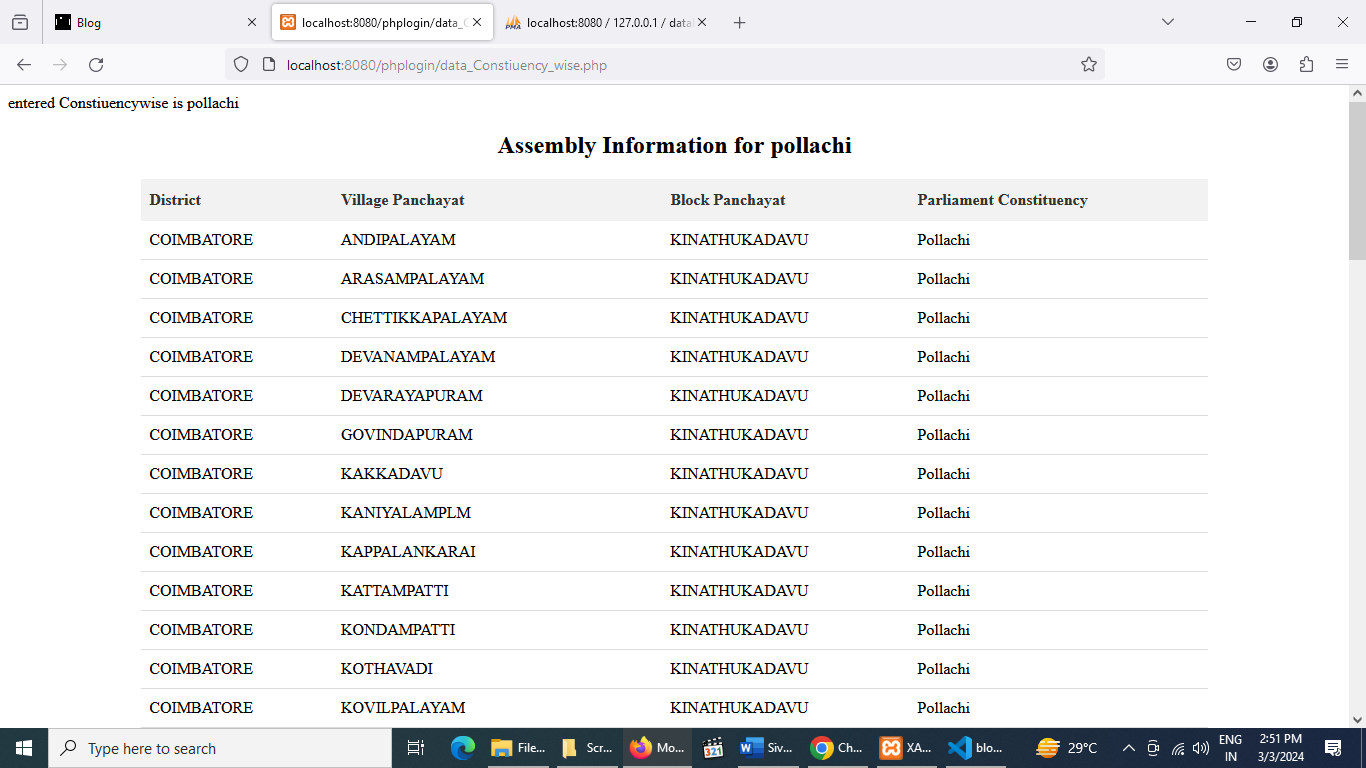
**All details**



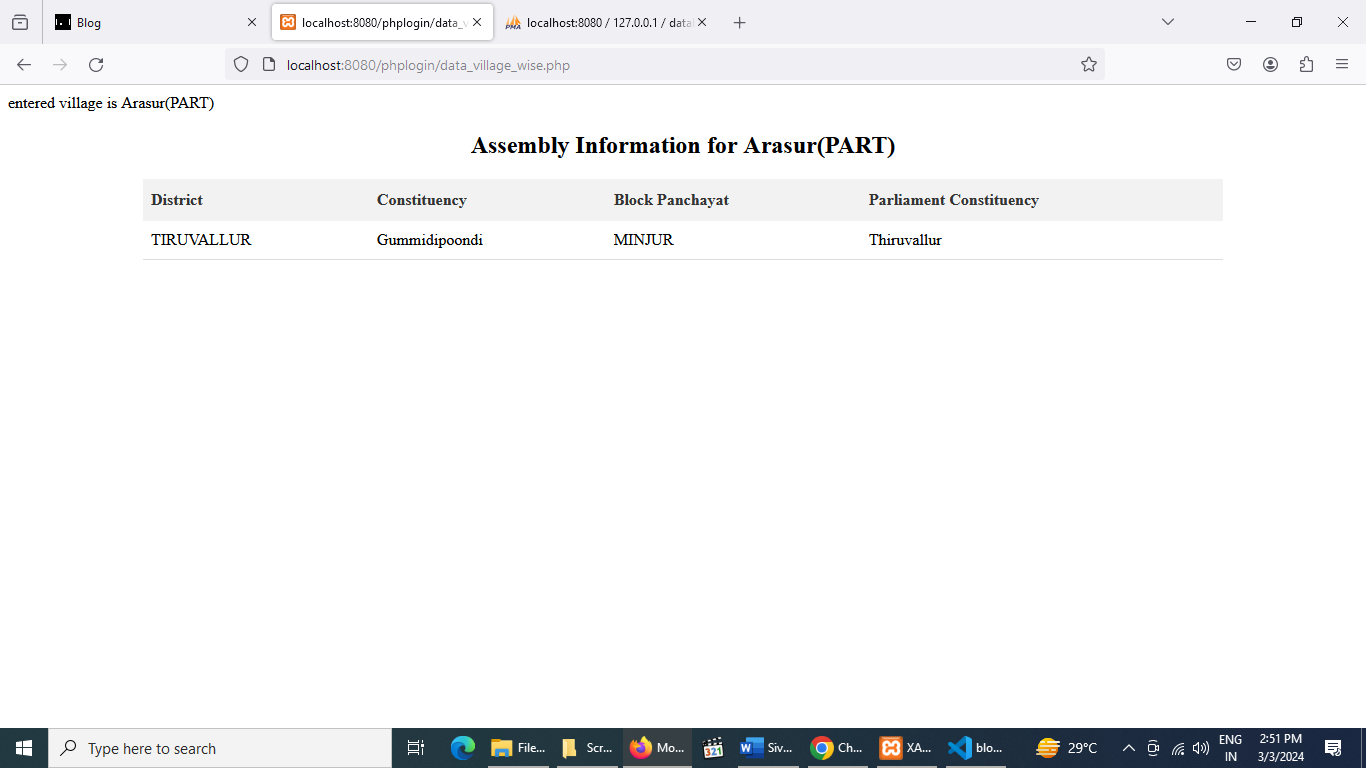
**District wise data**



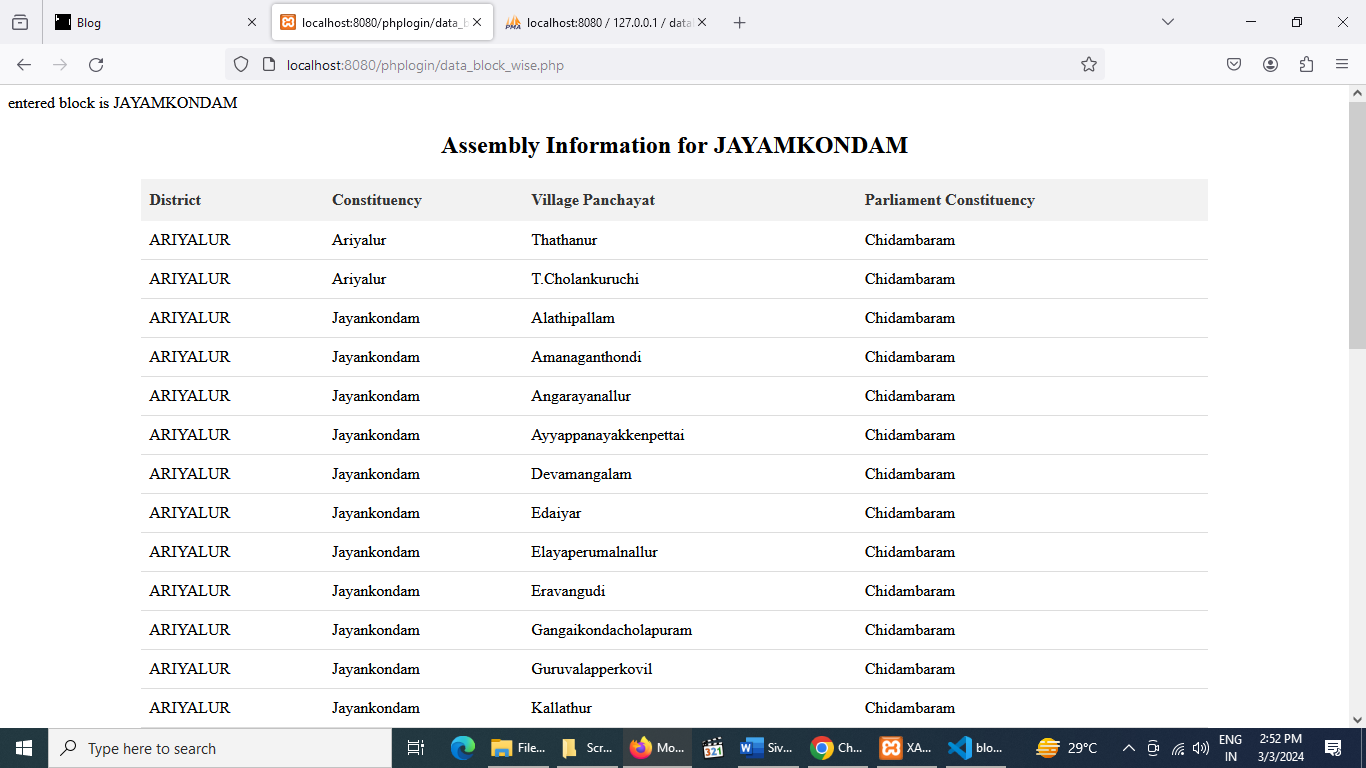
**Constiuency wise data**



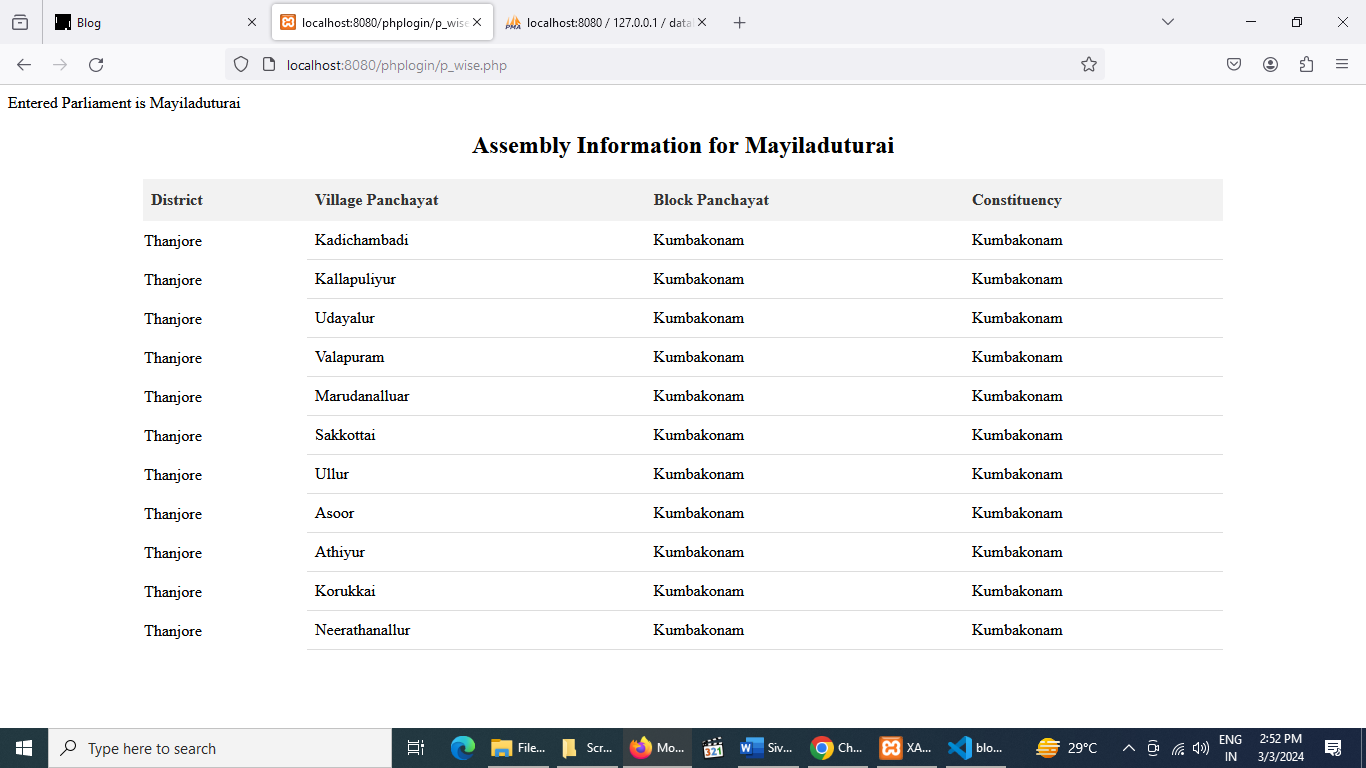
**Village wise data**



**Block wise data**



**Parliament wise data**

****

**CONCLUSION**

This project is only a humble venture to satisfy the needs to manage the required project work. Several user-friendly coding also adopted. This package shall prove to be a powerful package in satisfying all the requirements of the curriculum The objective of software planning is to provide a framework that enables the manager to make reasonable estimates made within a limited time frame at the beginning of the software project and should be updated regularly as the project progresses. The purpose of directory management system is to automate the existing manual system by the help of computerized and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. The required software and hardware are easily available and easy to work with. The beneficiaries can maintain computerized records without redundant entries. That means that one need not be distracted by information that is not relevant, while being able to reach the information.

**FUTURE ENHANCEMENTS**

The project has a very vast scope in future. The project can be implemented on intranet in future. Project can be updated in near future as and when requirement for the same arises, as it is very flexible in terms of expansion. With the proposed software of database Space Manager ready and fully functional the is now able to manage and hence run the entire work in a much better, accurate and error free manner. Can be added multilingual to this site. The project can be developed to encompass all districts, offering comprehensive directory services across a wide geographic area. Integrate GIS data to provide location-based services, such as mapping of government facilities, boundary information, and service areas. Obtain gender-disaggregated population data from reliable sources such as national census reports, demographic surveys, or local government records. Ensure that the data is regularly updated to reflect changing demographics.

**BIBLIOGRAPHY**

1. Larry Ullman, "PHP and MySQL for Dynamic Web Sites: Visual QuickPro Guide”, 5th Edition, 2017
2. Matt Zandstra, "PHP Objects, Patterns, and Practice", 5th Edition, 2004
3. Jon Duckett. " HTML and CSS: Design and Build Websites”, 1st Edition2011.
4. Jennifer Niederst Robbins. "Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics.", 5th Edition, 2018
5. Bennatan, TMH., “Software Project Management- A Practitioners Approach.”, 5th Edition, 1995

**WEBSITE REFERENCES**

1. https://www.tutorialspoint.com
2. [https://www.javatpoint.com](https://www.javatpoint.com/)
3. [https://www.geeksforgeeks.org](https://www.geeksforgeeks.org/autocmplete-combobox-in-python-tkinter/)
4. <https://copyprogramming.com>
5. <https://stackoverflow.com>
6. <https://amankharwal.medium.com>
7. <https://www.datacamp.com>