**Chapter – 1**

**INTRODUCTION**

* 1. **ABOUT THE SRI SHYLA VIDYA SAMASTHE**

SRI SHYLA VIDYA SAMASTHE was established in 1992 and it is managed by the Pvt. Unaided. It is located in Urban area. It is located in SOUTH1 block of BENGALURU U SOUTH district of Karnataka. The school consists of Grades from 1 to 10. The school is Co-educational and it doesn't have an attached pre-primary section. The school is N/A in nature and is not using school building as a shift-school. Kannada is the medium of instructions in this school. This school is approachable by all weather road. In this school academic session starts in April.  
         The school has Private building. It has got 8 classrooms for instructional purposes. All the classrooms are in good condition. It has 2 other rooms for non-teaching activities. The school has a separate room for Head master/Teacher. The school has Pucca boundary wall. The school has have electric connection. The source of Drinking Water in the school is Tap Water and it is functional. The school has 1 boys toilet and it is functional. and 1 girls toilet and it is functional. The school has a playground. The school has a library and has 300 books in its library. The school does not need ramp for disabled children to access classrooms. The school has 10 computers for teaching and learning purposes and all are functional. The school is having a computer aided learning lab. The school is Not Provided providing mid-day meal.

The School has been recognized by Council for Indian School Certificate Examinations since a long time. This School has been viewed 195 times by the visitors on CBSE. This School is counted among the top-rated Schools in Karnataka with an excellent academic track record. If you're looking for more details regarding results, examinations schedule, admission procedure, application forms and syllabus, kindly contact the relevant department of the school.

This school is looking for teachers with qualifications such as B.Ed, D.Ed, TTC, NTT, or Montessori. They are looking for teachers for positions such as mathematics, French, English, chemistry, physics, pre-primary, physical education, and art and craft. Applicants with experience are preferred  
This school offers a Coaching For School Success (CFSS) program to help teachers improve their teaching skills. The program focuses on building a collaborative culture among teachers, and emphasizes the importance of diagnosing gaps in the teaching-learning process.

* 1. **ABOUT THE PROJECT**

The **Personalized Learning Platform** is an innovative web-based application designed to streamline the process of sharing and accessing academic materials between teachers and students. In the modern educational environment, there is a growing need for centralized and efficient systems that enable seamless communication and resource distribution. This project addresses these needs by providing a dedicated platform for uploading, organizing, and accessing subject-specific learning materials in an intuitive and secure manner.

The platform is primarily aimed at bridging the gap between teachers and students by creating a digital repository where academic resources can be stored and accessed with ease. Teachers often face challenges in distributing study materials effectively, while students may struggle with scattered or inconsistent access to essential resources. By addressing these pain points, the Personalized Learning Platform ensures that educational materials are always readily available, well-organized, and accessible to the intended audience.

At its core, the platform focuses on two primary user groups: teachers and students. Teachers can log in using their unique credentials, such as their name and ID, to upload subject-specific notes. These materials are then categorized and stored systematically in the platform's database. On the other hand, students can also log in using their respective IDs and names to browse through available resources and download notes as per their academic requirements. The system's user-friendly interface and structured organization ensure that even non-technical users can navigate it effortlessly.

The development of this platform stems from a clear need to modernize the way academic materials are shared. Traditional methods of distributing notes—such as email or physical handouts—are not only time-consuming but also lack consistency and accessibility. A centralized digital platform overcomes these limitations by providing a secure and always-available system where materials are stored by subject and associated with the respective teachers. Students no longer need to rely on external communication or ad-hoc methods to access critical study materials; they can simply log in and retrieve what they need.

The Personalized Learning Platform is also designed with security and data privacy in mind. Teachers and students have separate login portals, ensuring that only authorized individuals can access specific features of the platform. Teachers can upload resources without worrying about unauthorized downloads, while students are assured of accessing authentic and accurate materials directly from their educators. By integrating these features, the platform not only enhances usability but also builds trust among its users.

In summary, the Personalized Learning Platform provides a comprehensive solution to the challenges of resource sharing in education. It simplifies the interaction between teachers and students, organizes materials by subject for easy access, and ensures that resources are always available to those who need them. By adopting this platform, educational institutions can significantly improve the learning experience, fostering an environment of efficiency, accessibility, and collaboration.

* 1. **SOFTWARE DEVELOPMENT METHODOLOGY**

The development of the Personalized Learning Platform relied on a combination of robust software tools and technologies to ensure its functionality, scalability, and user-friendliness. These are the software we used in the project:

**Visual Studio Code (VS Code):**

VS Code served as the primary development environment for writing and managing the codebase. It was used to write the frontend code (HTML, CSS, JavaScript) and backend code (PHP). The integrated terminal allowed seamless execution of commands, while extensions for PHP and MySQL enhanced code efficiency and debugging. Code and version control through Git were also managed within VS Code.

**XAMPP:**

XAMPP was utilized to set up a local server environment for the project. It provided the Apache server and MySQL database, which allowed us to test the platform locally before deploying it. The control panel made it easy to manage services such as starting and stopping the server, ensuring smooth integration of backend logic with the database.

**MySQL:**

MySQL was used as the database management system to store and organize user data, including teacher and student login credentials, uploaded notes, and subject information. SQL queries were written to create, retrieve, update, and delete data efficiently. The database schema was designed to ensure that the notes were categorized by subject and associated with the respective teachers and students.

**PHP:**

PHP was the primary server-side scripting language used to handle backend operations. It was responsible for processing user login requests, managing file uploads, interacting with the MySQL database, and dynamically generating content for the frontend. For example, when a teacher uploaded notes, PHP scripts stored the files on the server and updated the database with relevant metadata.

**HTML, CSS, and JavaScript:**

HTML was used to structure the web pages, defining the layout for the login forms, dashboards, and file upload/download sections. CSS was applied to style these pages, ensuring a visually appealing and responsive user interface. JavaScript was employed to enhance interactivity, such as validating user inputs during login and providing real-time feedback for file uploads.

By using these tools and technologies in an integrated manner, the project was successfully developed to meet its objectives. Each software component played a specific role in ensuring that the platform was robust, user-friendly, and capable of handling the needs of teachers and students efficiently.

**Chapter – 2**

**REQUIREMENT SPECIFICATION**

* 1. **FUNCTIONAL REQUIREMENTS**

1. User Authentication:
2. File Upload
3. File Download
4. Data Organization
5. Dashboard Interface
   * 1. **FR – 1**

User Authentication :The platform must provide separate login systems for teachers and students, requiring valid credentials for access.

* + 1. **FR – 2**

File Upload : Teachers should be able to upload academic materials (e.g., notes, PDFs) to the platform, categorized by subject.

* + 1. **FR – 3**

File Download **:** Students must be able to browse and download notes organized by subject from the platform.

* + 1. **FR – 4**

Data Organization:Uploaded files must be stored systematically in the database, ensuring easy retrieval and access.

* + 1. **FR – 5**

Dashboard Interface:Teachers and students should have access to personalized dashboards displaying relevant information such as uploaded or available notes.

* 1. **NON-FUNCTIONAL REQUIREMENTS**

**Performance**

**Reliability**

**Scalability**

**Security**

**Usability**

* + 1. **NR – 1**

Performance **:** The platform must respond to user actions (e.g., login, file uploads/downloads) within a few seconds to provide a seamless experience.

* + 1. **NR – 2**

Reliabilit**y :** The system should be highly reliable, with minimal downtime and robust error-handling mechanisms to ensure continuous availability.

* + 1. **NR – 3**

Scalability : The platform must be capable of handling an increasing number of users and larger amounts of data as the user base grows.

* + 1. **NR – 4**

Security :User data, including login credentials and uploaded files, must be securely encrypted to protect against unauthorized access and breaches.

* + 1. **NR – 5**

Usability:The interface should be intuitive and easy to navigate for users with varying levels of technical expertise.

**Chapter – 3**

**REQUIREMENTS MODELLING**

Briefly describe the requirements modelling

* 1. **SCENARIO BASED MODELLING**

Briefly describe the Scenario based modelling

* + 1. **USE CASE DIAGRAM**

Use case diagram - figure with description

* + 1. **ACTIVITY DIAGRAM**

Activity diagram - figure with description

* + 1. **SWIMLANE DIAGRAM**

Swimlane diagram - figure with description

* 1. **DATA MODELLING**

Briefly describe the Data modelling

* + 1. **ENTITY RELATIONSHIP DIAGRAM**

ER diagram - figure with description

* 1. **CLASS BASED MODELLING**

Briefly describe the Scenario based modelling.

* + 1. **CLASS DIAGRAM**

Class diagram - figure with description

* + 1. **CRC CARDS**

CRC Cards - figure with description

**Chapter – 4**

**IMPLEMENTATION**

* 1. **SYSTEM ARCHITECTURE**

For this project, Scenario-Based Modelling identifies and documents how users (Administrator, Teacher, and Student) interact with the personalized learning platform. The approach ensures the design supports realistic use cases and addresses user requirements effectively.

1. Administrator Role:

o The Administrator manages the system by overseeing user registration and monitoring the uploaded content.

o Scenarios:

 Register and authenticate as an administrator.

 Add, edit, or delete users (teachers and students).

 Review uploaded materials and generate reports for platform usage.

2. Teacher Role:

o Teachers interact with the platform to upload subject-specific learning materials.

o Scenarios:

 Register and log in as a teacher.

 Select subjects they are teaching.

 Upload notes, validate content, and ensure it is accessible to enrolled students.

3. Student Role:

o Students access learning materials uploaded by their respective teachers.

o Scenarios:

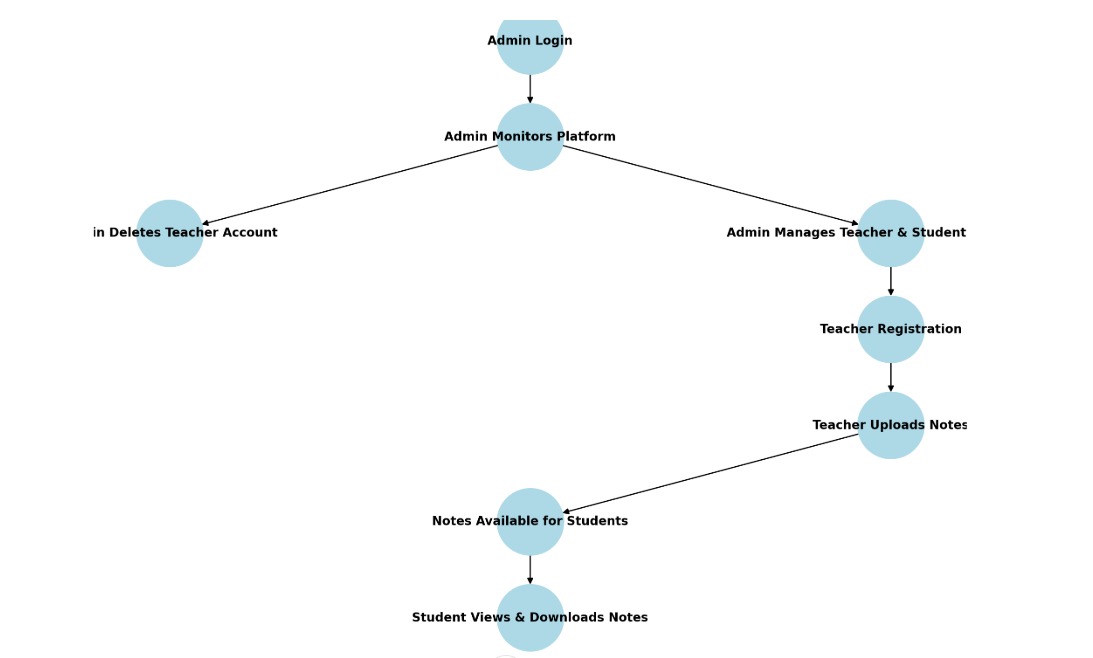
 Register and log in as a student.

 View and select enrolled subjects.

 Access and download subject-related notes for personalized learning.

By breaking down workflows for each user type, this modeling ensures the platform’s features align with user needs and system functionality.

* + 1. **FLOW CHART**

****

* 1. **TECH STACK**

The **Personalized Learning Platform** was developed using a robust and well-integrated tech stack to ensure seamless functionality, scalability, and a user-friendly experience. The chosen tools and technologies played a vital role in both the development and deployment of the platform. Below is a detailed explanation of the tech stack used:

#### **Frontend Technologies**

1. **HTML (HyperText Markup Language):**  
   HTML was used to structure the web pages of the platform. Each page, including the login forms, dashboards, and upload/download sections, was created using semantic HTML elements to ensure accessibility and maintainability. HTML forms were used for user input, such as logging in, uploading files, and navigating through the platform.
2. **CSS (Cascading Style Sheets):**  
   CSS was employed to style the platform's web pages, ensuring a clean and professional appearance. The design emphasized responsiveness, so the platform would work smoothly on different devices, including desktops, tablets, and smartphones. Custom styles were written to enhance visual aesthetics, such as buttons, tables, and file upload/download interfaces, while maintaining a consistent theme across the platform.
3. **JavaScript:**  
   JavaScript was used to add interactivity and improve user experience. For instance, JavaScript validated user input in real-time during login and file upload processes. Dynamic updates, such as showing success or error messages without refreshing the page, were also implemented using JavaScript.

#### **Backend Technologies**

1. **PHP (Hypertext Preprocessor):**  
   PHP served as the primary server-side scripting language to manage backend operations. It was responsible for processing user requests, interacting with the database, and generating dynamic content. Key tasks performed using PHP include:
   * Authenticating users during login by verifying credentials against the database.
   * Handling file uploads, ensuring secure storage of academic materials on the server.
   * Fetching data from the database to display relevant information, such as available notes for a specific subject, on the dashboard

#### **Database Management System**

1. **MySQL:**  
   MySQL was used to manage and store all the data for the platform. It ensured efficient handling of:
   * **User Data:** Login credentials for teachers and students, stored securely to facilitate authentication.
   * **Subject Information:** A catalog of subjects, enabling proper categorization of uploaded notes.
   * **Uploaded Notes:** File metadata such as file name, associated teacher, subject, and timestamps.

SQL queries were written to perform CRUD (Create, Read, Update, Delete) operations effectively. Database schema was designed to maintain relationships between users, subjects, and uploaded files, ensuring logical data organization.

#### **Local Development Environment**

1. **XAMPP:**  
   XAMPP was used to set up a local development environment. It provided essential services such as:
   * **Apache Web Server:** Enabled local hosting for testing the web application.
   * **MySQL Server:** Supported the database management system for storing and retrieving data.  
     The XAMPP control panel simplified the process of managing server operations like starting/stopping Apache and MySQL during development.

**Code Editor**

1. **Visual Studio Code (VS Code):**  
   VS Code was the primary development environment used for writing and managing the codebase. Its features included:
   * Syntax highlighting and code linting for PHP, HTML, CSS, and JavaScript.
   * Integrated terminal for running commands and debugging code efficiently.
   * Extensions for PHP and MySQL, which improved code readability and error detection.
   * Git integration for version control, allowing collaborative development and code backups.
   1. **SAMPLE CODE**

**Login page for Students/Teachers**

<?php

session\_start();

include('db.php');  // Include your DB connection

// Handle the login process

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

    $name = $\_POST['name'];

    $email = $\_POST['email'];

    $password = $\_POST['password'];

    // Check if the user is a student

    $stmt = $conn->prepare("SELECT \* FROM students WHERE name = ? AND email = ? AND password = ?");

    $stmt->bind\_param("sss", $name, $email, $password);

    $stmt->execute();

    $result = $stmt->get\_result();

    if ($result->num\_rows > 0) {

        $student = $result->fetch\_assoc();

        $\_SESSION['student\_id'] = $student['id'];  // Store student ID in session

        $\_SESSION['name'] = $student['name'];      // Store student name in session

        header("Location: student\_dashboard.php");  // Redirect to student dashboard

        exit();

    } else {

        // Check if the user is a teacher

        $stmt = $conn->prepare("SELECT \* FROM teachers WHERE name = ? AND email = ? AND password = ?");

        $stmt->bind\_param("sss", $name, $email, $password);

        $stmt->execute();

        $result = $stmt->get\_result();

        if ($result->num\_rows > 0) {

            $teacher = $result->fetch\_assoc();

            $\_SESSION['teacher\_id'] = $teacher['id'];  // Store teacher ID in session

            $\_SESSION['name'] = $teacher['name'];      // Store teacher name in session

            header("Location: teacher\_dashboard.php");  // Redirect to teacher dashboard

            exit();

        } else {

            $message = "Invalid credentials!";

        }

    }

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <link rel="stylesheet" href="css/styles.css">

    <title>Login</title>

    <style>

    .footer {

            text-align: center;

            margin-top: 20px;

        }

        .footer .button {

            display: inline-block;

            padding: 10px 20px;

            background-color: #007bff;

            color: #fff;

            text-decoration: none;

            border-radius: 8px;

            font-size: 16px;

            text-align: center;

            box-shadow: 0 4px 6px rgba(0, 0, 0, 0.2);

            transition: all 0.3s ease;

        }

        .footer .button:hover {

            background-color: #0056b3;

        }

    </style>

</head>

<body>

    <div class="form-container">

        <h2>Login</h2>

        <?php if (isset($message)) { echo "<div class='message'>$message</div>"; } ?>

        <form method="POST" action="login.php">

            <label for="name">Name:</label>

            <input type="text" name="name" placeholder="Enter your name" required>

            <label for="email">Email:</label>

            <input type="email" name="email" placeholder="Enter your email" required>

            <label for="password">Password:</label>

            <input type="password" name="password" placeholder="Enter your password" required>

            <button type="submit" class="button">Login</button>

        </form>

        <p>Don't have an account? <a href="registration.php">Register here</a></p>

    </div>

      <!-- Add the Home Button in the footer -->

      <div class="footer">

        <a href="index.php" class="button">Home</a> <!-- This button will redirect to the home page -->

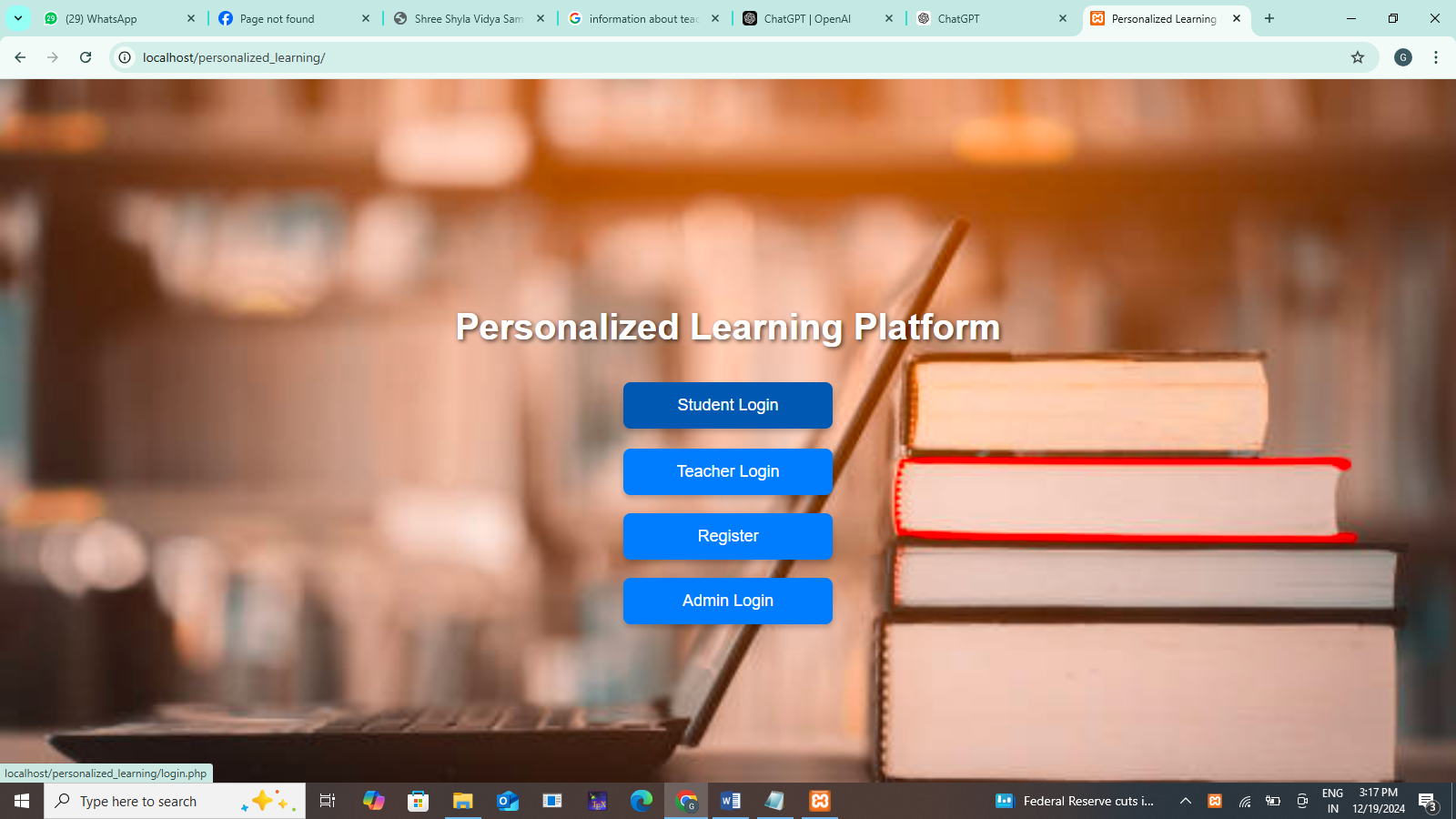
    </div>

</body>

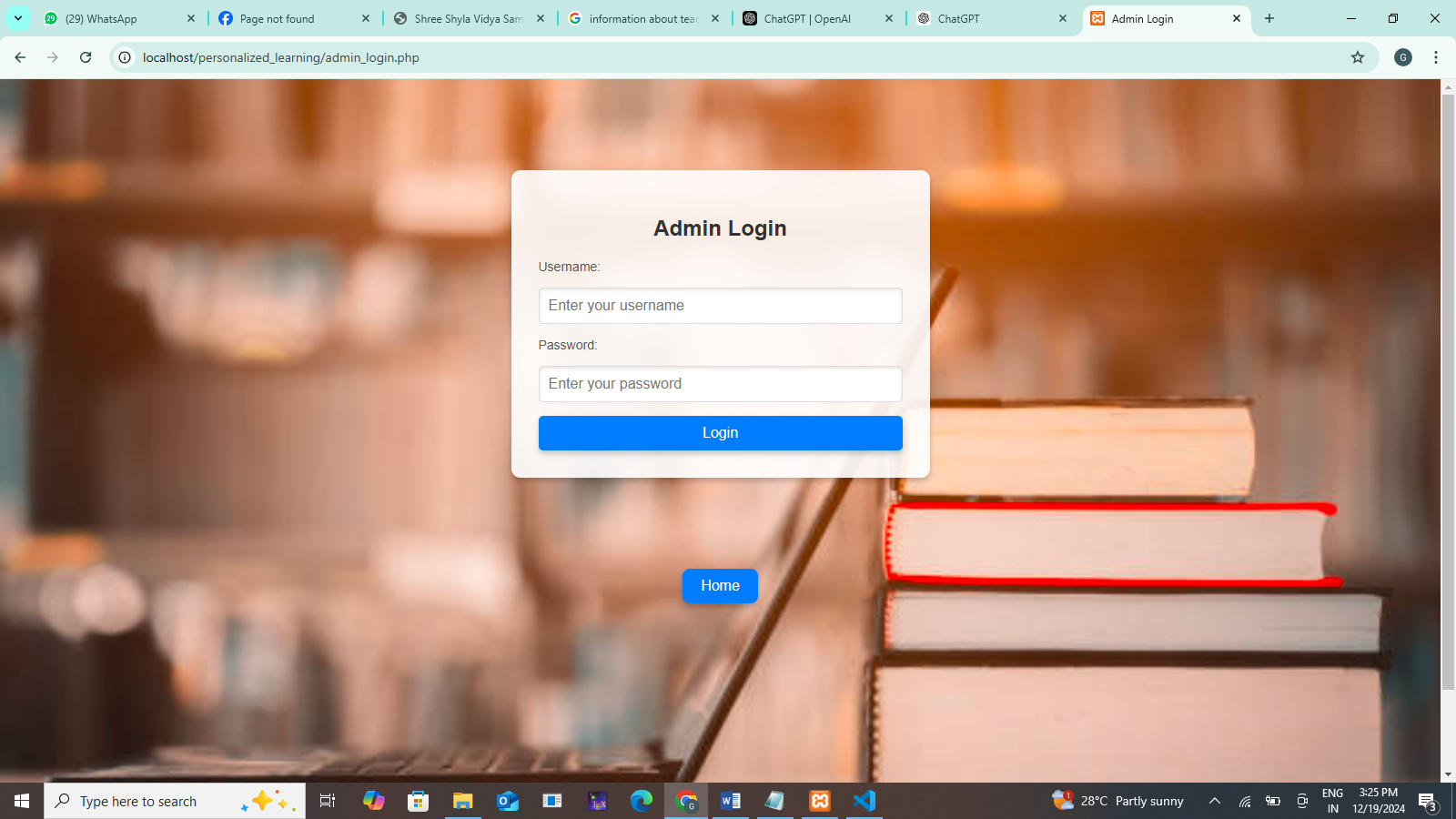
</html>

* 1. **OUTPUT SCREENSHOTS**

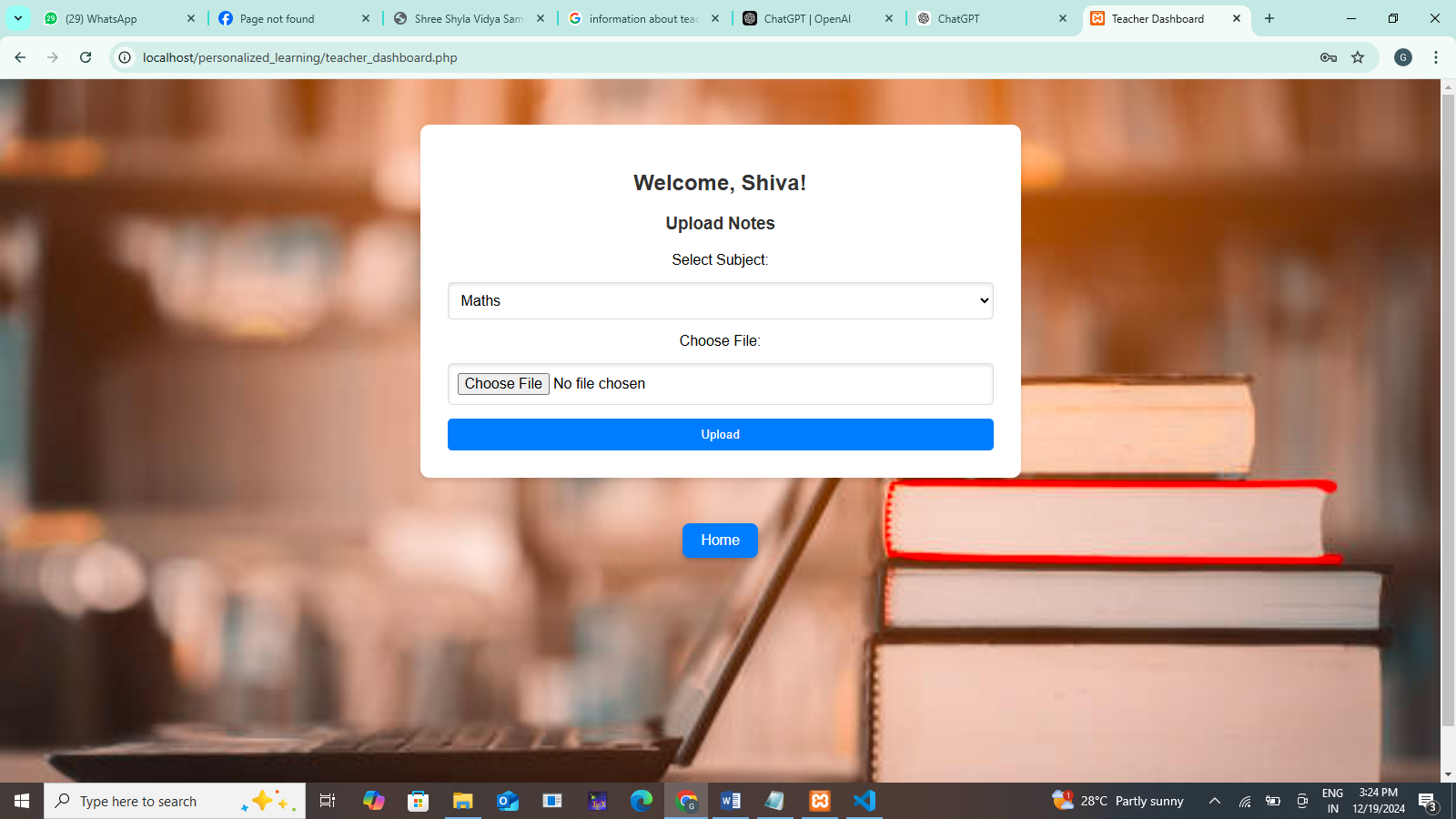
**Index page**

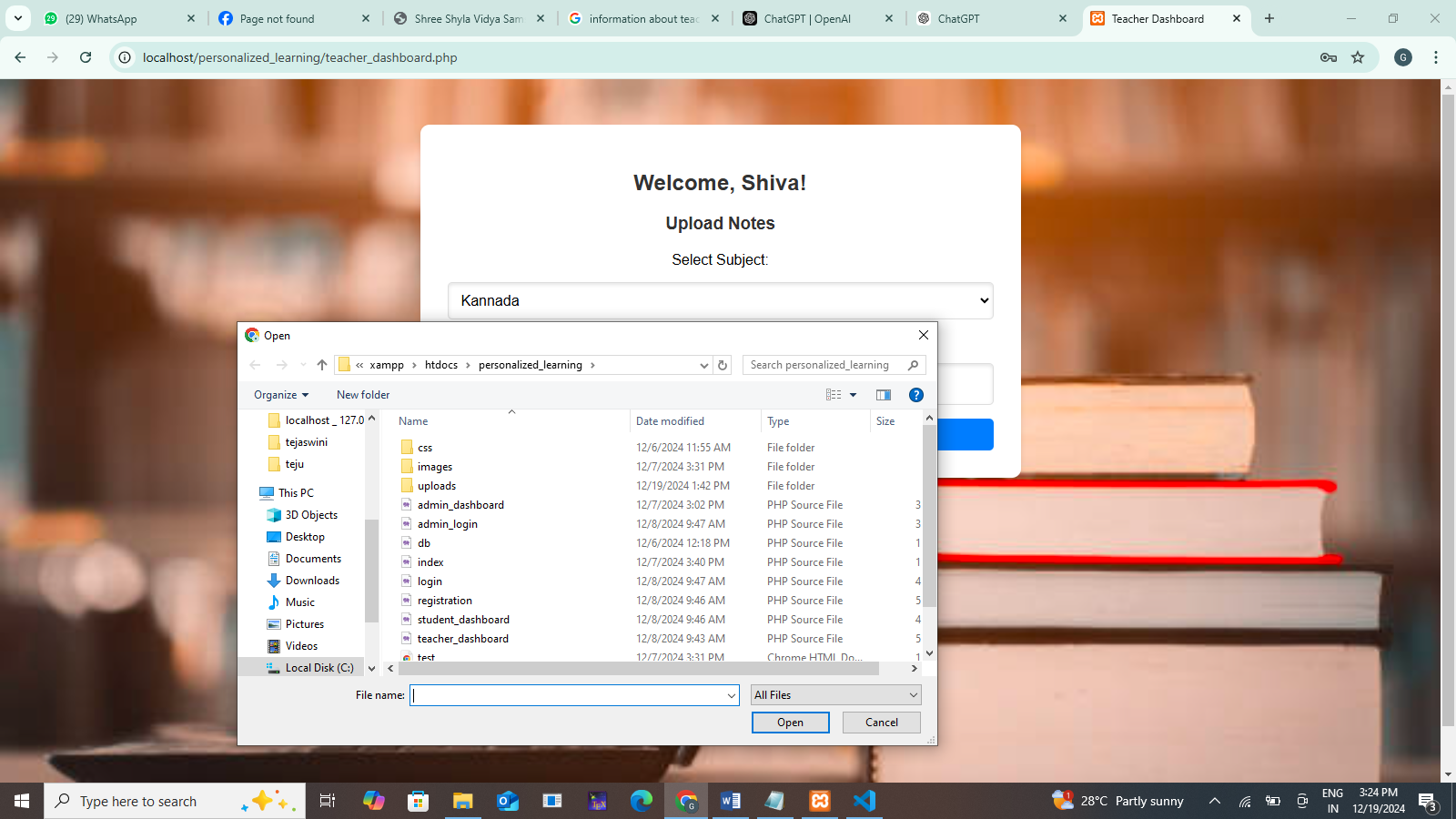


Admin login:

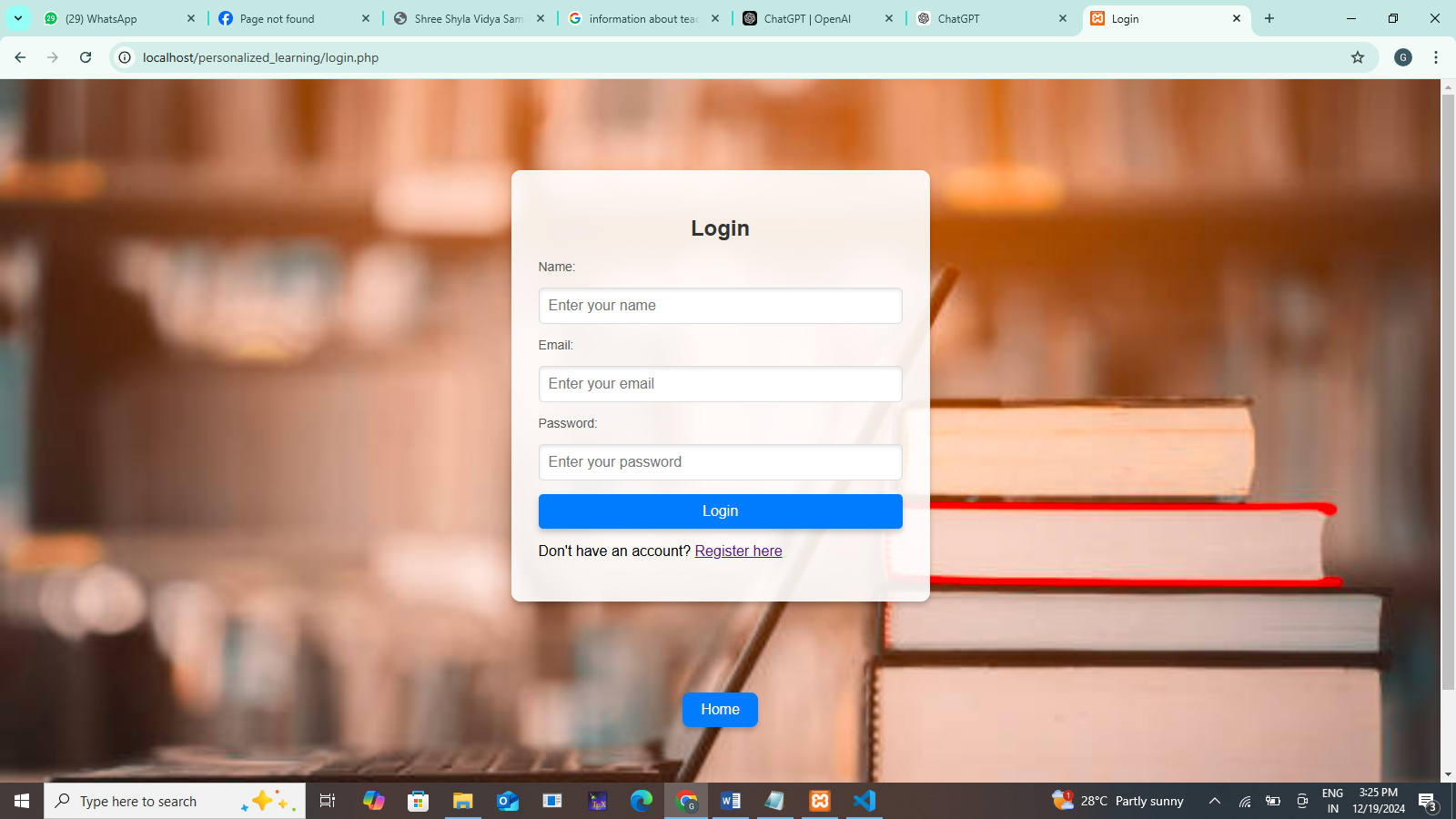


Teacher dashboard:



****

**Teacher/Student Login:**

****

**Chapter – 5**

**TESTING**

* 1. **UNIT TESTING**

 **Login System Testing:**

* Separate unit tests were created for both teacher and student login functionalities.
* Tests verified that:
  + Valid credentials (e.g., correct name and ID) granted access.
  + Invalid credentials displayed appropriate error messages.
  + Edge cases such as empty fields, special characters, and overly long inputs were handled securely to prevent system crashes or unauthorized access.

 **File Upload Functionality:**

* Tests were conducted to ensure teachers could successfully upload files (e.g., PDFs, Word documents).
* The system was validated to:
  + Accept only specific file types and reject invalid formats.
  + Handle file size limits, ensuring large files were not uploaded.
  + Display error messages for failed uploads.

 **File Download Functionality:**

* Unit tests confirmed that students could access and download files associated with their subjects.
* The tests ensured that:
  + Only authorized students could download files.
  + Corrupt or incomplete downloads were handled gracefully.
  + Files were accurately retrieved from the database and server.

 **Database Operations:**

* CRUD (Create, Read, Update, Delete) operations were tested for accuracy:
  + Creating new user accounts and notes entries in the database.
  + Retrieving uploaded notes and associating them with the correct subjects.
  + Updating records such as user credentials or file metadata.
  + Deleting obsolete or duplicate entries from the database without affecting other records.

 **Dashboard Interface Testing:**

* Unit tests ensured that the dashboards for teachers and students displayed accurate and relevant information.
* The responsiveness and functionality of search and filter options were validated.
  1. **INTEGRATION TESTING**

 **Login System and Dashboard:**

* Verified the interaction between the login module and the dashboard.
* Ensured that:
  + Teachers were redirected to their personalized dashboards upon successful login.
  + Students were redirected to their respective dashboards based on their login credentials.
  + Invalid login attempts did not grant access to the dashboard.

 **File Upload and Database Integration:**

* Tested the integration between the file upload functionality and the database.
* Ensured that:
  + Uploaded files were stored correctly in the server directory.
  + File metadata (e.g., file name, subject, uploader) was accurately recorded in the MySQL database.
  + The database and server maintained synchronization when files were uploaded or deleted.

 **File Download and Database Interaction:**

* Verified that students could download files associated with their subjects.
* Ensured the system retrieved file details from the database correctly and directed the student to the correct file location on the server.
* Tested for scenarios where files were missing or incorrectly linked in the database.

 **Search and Filter Functionality:**

* Tested the integration between the search and filter features and the database.
* Ensured that:
  + Search results accurately reflected notes stored in the database.
  + Filters worked correctly to display subject-specific files.
  + No performance issues occurred when querying the database for large datasets.

 **User Management and Access Control:**

* Verified that access control mechanisms were functioning as expected:
  + Teachers could upload and manage only their files.
  + Students could access and download files but not upload or delete files.
* Checked role-based permissions and ensured no unauthorized access to restricted areas.
  1. **SYSTEM TESTING**

1. **Preparation:**
   * A comprehensive test plan was developed based on the functional and non-functional requirements.
   * Test cases were written to cover all aspects of the platform, including user interactions, backend operations, and overall performance.
2. **Testing Scenarios:**
   * User Authentication:
     + Verified that teachers and students could log in with valid credentials.
     + Tested error handling for invalid credentials, empty fields, and SQL injection attempts.
   * File Upload and Download:
     + Ensured teachers could upload files, which were correctly stored in the database and accessible for students.
     + Verified that students could only download files relevant to their subjects.
   * Dashboard Navigation**:**
     + Tested that users were directed to the correct dashboard after logging in.
     + Ensured dashboards displayed relevant information, such as uploaded notes for teachers and available resources for students.
   * Search and Filter:
     + Validated the functionality of search and filter options to ensure notes were retrieved accurately based on user queries.
   * Role-based Access Control:
     + Confirmed that only authorized users could access restricted areas or perform specific actions, such as uploading files.
3. **Non-Functional Testing:**
   * Performance Testing:
     + Evaluated the system’s responsiveness and loading times with multiple simultaneous users accessing and uploading files.
   * Security Testing:
     + Verified data encryption for user credentials and ensured sensitive data (e.g., notes) was protected from unauthorized access.
   * Compatibility Testing:
     + Tested the platform on different browsers (e.g., Chrome, Firefox, Edge) and devices (e.g., desktops, tablets, smartphones) to ensure a consistent user experience.
4. **Test Environment:**
   * The system was tested in a controlled environment using XAMPP to simulate the server and database operations.
   * Real-world scenarios were replicated using mock data for user accounts, subjects, and files.
5. **Defect Identification and Resolution:**
   * Detected and resolved issues such as broken links, incorrect file associations, and inconsistent dashboard displays.
   * Ensured that error messages were user-friendly and provided clear instructions for resolution

**Chapter – 6**

**CONCLUSION AND FUTURE ENHANCEMENTS**

* 1. **CONCLUSION**

The **Personalized Learning Platform** has been successfully developed as a comprehensive solution to bridge the gap between teachers and students in terms of resource sharing and academic material management. The platform enables teachers to efficiently upload and organize subject-specific notes, while students can easily access and download these resources through an intuitive and secure interface. By leveraging modern technologies such as PHP, MySQL, and a responsive design, the platform ensures accessibility, scalability, and reliability across various devices and user scenarios.

The project achieved its primary objectives, including simplifying the resource-sharing process, enhancing user experience through a structured dashboard, and maintaining data security with robust authentication mechanisms. The adoption of the platform is expected to streamline academic workflows, foster collaboration between educators and learners, and ensure consistent access to high-quality learning materials.

* 1. **FUTURE ENHANCEMENTS**

**** Real-Time Communication**:** Integrating a chat or messaging system to enable direct communication between teachers and students for clarifications or additional guidance.

 Analytics Dashboard: Adding analytics to provide insights into student engagement, such as tracking downloads, most accessed notes, and study patterns.

 Mobile Application**:** Developing a dedicated mobile application to provide an optimized experience for smartphone users and enable offline access to downloaded notes.

 Multi-Language Support **:**Incorporating multilingual support to cater to a diverse audience, especially in regions with multiple languages of instruction.