Appointment Scheduler

ABIN SAJI

BEVEN K BINOY

VISHNU PRASAD

Department of Computer Applications

SAINTGITS COLLEGE OF ENGINEERING (Autonomous)

Kottukulam Hills, Pathamuttom P.O., Kottayam 686532

November 2024

ABSTRACT

This project presents a **web-based appointment scheduling system** developed in PHP, designed to optimize the management and scheduling of appointments for healthcare providers. The application replaces traditional manual booking systems with an automated, user-friendly, and accessible platform. It provides a seamless experience for both clients and healthcare providers, ensuring efficient scheduling and management.

The system offers functionalities like **booking**, **rescheduling**, **cancelling appointments**, and user authentication to ensure secure access. The platform includes a **client dashboard** where clients can manage their appointments, including accepting or cancelling bookings. Healthcare providers and administrators will have a dedicated **admin dashboard** where they can set available time slots, manage appointments, and view daily, weekly, or monthly schedules.

With a **PHP backend** that integrates with a **MySQL database**, the system securely stores user data, appointment details, and availability schedules. **JavaScript** enhances interactivity, and the **UI** is designed to be responsive and intuitive using **HTML and CSS**.

This system is scalable, allowing integration with third-party calendar services and supports multiple user roles. It is adaptable to various appointment-based services, offering a robust and versatile solution.

Key Modules:

1. **User Authentication Module:**
   - Users (healthcare providers, administrators) and clients must register and log in securely.
   - Separate registration and login systems for healthcare providers/admins and clients.
   - Secure password hashing, login verification, and session management.
   - Role-based access control (admin, provider, client).
2. **Client Registration & Dashboard Module:**
   - **Client Registration**: Clients can register for an account with basic personal details (name, email, phone number).

- **Client Dashboard**: After logging in, clients can:
  - **View Upcoming Appointments**: A list of scheduled appointments.
  - **Accept or Cancel Appointments**: Clients can manage their bookings by accepting or cancelling appointments directly from their dashboard.
  - **Update Profile**: Edit personal information, contact details, and preferences.
- **Notification**: Clients will receive email/SMS notifications for confirmed, pending, or cancelled appointments.

3. **User (Admin/Healthcare Provider) Registration & Dashboard Module:**
   - **User Registration**: Healthcare providers/admins can register and set up their professional profile (e.g., specialization, clinic details).
   - **Admin Dashboard**:
     - Manage users, view and edit appointments, manage system settings.
     - Manage time slots and availability.
     - View analytics, reports, and appointment trends.
   - **Provider Dashboard**:
     - View scheduled appointments.
     - Manage available time slots.
     - View clients' appointment history.
4. **Appointment Booking Module:**
   - **Clients** can browse available time slots based on the provider's calendar and book appointments.
   - **Confirmation of Booking**: Once a client books an appointment, a confirmation email/SMS is sent to both the client and the provider.
   - **Rescheduling**: Clients or providers can request changes to the appointment (admin approval may be required for rescheduling).
5. **Admin Dashboard Module:**
   - **View & Manage Appointments**: Admins can view all bookings, and manage them as needed.
   - **User Management**: Admins can create, update, and delete user profiles (both clients and healthcare providers).
   - **Analytics & Reporting**: Admins can generate reports for appointment trends, cancellations, and overall system usage.
6. **Calendar & Time Slot Management Module:**
   - **Providers can define their availability**: Providers can set their working hours and available time slots.
   - **Admin Can Set Provider Schedules**: Admin can help set provider availability or modify provider hours.
   - **Sync with Third-Party Calendars**: Future enhancements could include integration with Google Calendar or similar services.
7. **Notifications & Reminders Module:**
   - **Appointment Reminders**: Automated email/SMS notifications are sent to clients and providers as reminders for upcoming appointments.
   - **Appointment Confirmation**: Once an appointment is booked, both the client and provider receive a confirmation.
   - **Cancellation Alerts**: Clients and providers will receive immediate alerts if a booking is cancelled by either party.
8. **Appointment Management Module:**
   - **Provider View**: Providers can view and manage their upcoming appointments from their dashboard.
   - **Client View**: Clients can view their scheduled appointments, see if they are confirmed, pending, or cancelled, and manage their bookings directly.
   - **Cancel & Accept Bookings**: Clients can accept or cancel appointments based on provider

availability. The system will notify both the client and the provider.

9. **Analytics & Reporting Module:**
   - Generate reports about appointment trends, cancellation rates, and client demographics.
   - Providers and admins can use these insights to optimize their schedules and improve the overall service.
10. **Database Management Module:**
    - **Database Schema**: The MySQL database stores information such as user profiles, appointment records, time slots, and transaction histories.
    - **Data Security**: Secure database management practices, ensuring client and provider information is stored safely and in compliance with privacy regulations.

INTRODUCTION

1.1 Project Overview

This project focuses on developing a web-based “Appointment Scheduling system“ using PHP and MySQL to streamline the process of booking, managing, and tracking appointments for businesses and service providers. Designed with both user convenience and administrative ease in mind, the scheduler offers secure user authentication, real-time slot availability, and automated notifications via email or SMS to confirm and remind users of their appointments, reducing no-show rates and improving overall efficiency. Users can easily book, reschedule, or cancel appointments through a responsive, mobile-friendly interface, while administrators benefit from a centralized dashboard to configure availability, manage bookings, and access appointment statistics. The system includes role- based access control, ensuring that administrators have oversight and control while users access only their specific bookings. Utilizing AJAX for dynamic data updates, the platform provides a smooth, interactive experience without page reloads. By combining robust backend functionality with an intuitive frontend, the appointment scheduler minimizes the administrative burden of manual bookings and enhances user engagement. Planned future enhancements, such as third-party calendar integration, payment gateways, advanced analytics, and multi-language support, will further expand the system's capabilities, making it a versatile and scalable solution for a range of industries.

1.2 Organization Profile

SAINTGITS COLLEGE OF ENGINEERING (Autonomous) is a self- financing technical institution located at Kottayam district of Kerala. The college was established in 2002. Saintgits is approved by All India Council for Technical Education and affiliated to APJ Abdul Kalam Technological University, Kerala. The institute is accredited by NBA in 2016 for 3 years for 5 UG programs and in 2017 for 3 years for Master of Computer Application program. The college was founded by a group of well-known academicians. They are pioneering educators, having unmatched experience in the field of education with a belief that the continuous search for knowledge is the sole path to success. The primary focus of the institution is to expose the young minds to the world of technology, instilling in them confidence and fortitude to face new challenges that enable them to excel in their chosen fields. The college inculcates the development of all facets of the mind culminating in an intellectual and balanced personality. A team of dedicated and caring faculty strives to widen the student’s horizon of learning thereby achieving excellent results for every student. With a scientifically planned methodology combined with a team of handpicked faculty the best in the teaching profession and the state-of-the-art infrastructure, the quality of the engineering education at Saintgits is unparalleled in the region. The institute has turned into a benchmark for others to emulate. With 100% seats filled from the year of inception itself, and feel confident that Saintgits can serve even better with every passing year. Saintgits College of Engineering, right from inception, has been maintaining high levels of standards in academic and extra-curricular realms of activities. Saintgits offer a B Tech Degree course in 9 Engineering disciplines, and Masters's Degree courses in Engineering, Administration. In the short span of a decade of its existence and among the six batches of students that have graduated, the college bagged several universities ranks and has a remarkably high percentage of pass. The students of the first batch of MCA bagged the first two ranks in the University. The college is also the venue of national and state level seminars and symposiums and has emerged as the hub of technical education in Kerala.

REQUIRMENT AND ANALYSIS

2.1 Functional Requirements

The appointment scheduler system in PHP requires a comprehensive set of functional and non-functional features to meet user needs and performance standards. Functionally, it must provide secure user registration and authentication, including role-based access control for both clients and administrators. Users should be able to view available time slots, book, reschedule, or cancel appointments, with real-time updates to prevent double booking. Administrators, meanwhile, need the ability to manage time slots, configure blackout dates, and view all appointments through an interactive admin dashboard. Automated notifications via email or SMS will confirm appointments, send reminders, and inform users of cancellations, ensuring a streamlined booking experience. A calendar view displays upcoming appointments, with an admin option to view all bookings across different time periods for better oversight. To manage data, a robust MySQL database will securely store user and appointment details, ensuring data consistency and real-time updates during booking and modification actions.

2.2 Non-Functional Requirements

Non-functionally, the system must emphasize security, using encrypted data storage, secure session management, and parameterized queries to prevent SQL injection. Performance expectations include fast load times, real-time availability processing, and scalability to accommodate more users over time. The user interface must be intuitive and mobile-friendly to enhance usability, allowing for easy navigation and seamless booking on any device. Reliability is key, with aims for 99% uptime and reliable data handling to prevent loss or inconsistency during user interactions. For maintainability, modular code and detailed documentation will facilitate future updates and troubleshooting, supported by error logging for identifying issues. Technical feasibility of the system is high, with PHP and MySQL providing a solid foundation for web functionality, and AJAX enabling real-time updates to improve user experience. Security measures, like session management and encryption, will address common threats, while email and SMS notifications are enabled through dependable APIs. Server and browser compatibility, as well as API dependencies for notifications, are additional constraints that will ensure consistent operation across platforms. Together, these requirements and considerations create a cohesive, efficient system aligned with user and business needs.

2.3 Analysis

User Needs Analysis

Users need an accessible, user-friendly interface that allows them to easily view, book, reschedule, or cancel appointments. They require confirmation and reminder notifications to avoid missed appointments. Administrators need control over availability, viewing all appointments, and managing the system's day-to-day functioning without technical difficulty.

Technical Feasibility

PHP and MySQL offer a solid foundation for developing a web-based appointment scheduler with reliable database capabilities. The inclusion of AJAX for real-time updates improves interactivity, while tools like SMTP or third-party SMS APIs enable automated notifications. Security protocols will address potential threats, including secure session management and data encryption.

SYSTEM CONFIGURATION

3.1 Hardware Specification

The hardware specifications aim to support smooth operation for the application development, deployment, and usage. This will be particularly useful if the system will be deployed on a local machine, client computers, or a server.

Processor:
- **Minimum**: Dual-core processor at 2.0 GHz or higher.
- **Recommended**: Quad-core or higher with a clock speed of 2.5 GHz for better responsiveness, especially if the system will handle a lot of concurrent users or tasks.

RAM:
- **Minimum**: 4 GB DDR4.
- **Recommended**: 8 GB or more if the system will handle multiple users or use heavy operations like report generation, multi-tasking, or data analytics.

Hard Disk Space:

- **Minimum**: 2 GB free hard disk space.
- **Recommended**: 10 GB or more for storing system files, user data, logs, and backups.

SSD storage is preferable for faster data retrieval and system performance.

Input Device:

- Mouse and standard keyboard are required.
- **Additional**: Webcam for virtual appointments or video integration (Webcam 2.0 or higher).

Output Device:
- **Minimum**: Monitor with at least 1280 x 720 resolution.
- **Recommended**: Full HD (1920x1080) or higher resolution for better UI/UX experience.
-

3.2 Software Specification

The software setup is critical to ensure proper functioning and the ability to develop and run the appointment scheduler system smoothly.

Operating System:
• **Minimum**: Windows 7 or above.
Recommended: Windows 10 or 11 for better security, performance, and support for modern tools. Linux (Ubuntu) or macOS are also viable options for server-side deployment or development environments

Drivers/Packages:

- **PHP**: Used for backend processing, enabling server-side scripting and dynamic content generation.
- **Web Server**: Apache HTTP Server to serve web applications, handle HTTP requests, and integrate backend and frontend components

Development Tools:

- **HTML, CSS**: Used for frontend development to create the user interface and manage the layout, design, and styling of the web application.

- **phpMyAdmin**: A web-based tool for managing MySQL databases, allowing for easy creation, modification, and administration of databases.

Additional Software for Development and Deployment:

- **Web Browser**: Google Chrome, Mozilla Firefox, or Microsoft Edge for testing and viewing the application during development and deployment.
- **Database Management Tools**:
    - **phpMyAdmin**: Provides a user-friendly interface for managing and interacting with MySQL databases directly from a web browser.

3.3 Recommended Network & Security Configuration

For an appointment scheduler system, especially if it's intended for use by multiple users over a network or on the internet, the following network and security measures should be considered:

Network Requirements:

• **Internet Connection**: Required if the system is to be accessed remotely or if the database is hosted on a cloud server.
• **Bandwidth**: Minimum of 1 Mbps for small systems, and 10 Mbps or more for higher volumes of traffic.
• **Local Network**: A local area network (LAN) with sufficient bandwidth (100 Mbps or higher) if using the system internally.

Security Measures:

- **Firewall**: Configure firewalls to protect the system from unauthorized access.
- **SSL/TLS**: Secure Sockets Layer for encrypting communications between users and the server, especially if sensitive data (such as personal information or appointments) is transmitted.
- **Backup and Recovery**: Regular data backups of the MySQL database and server files to ensure data safety in case of failure.
- **Authentication & Authorization**: Secure login mechanisms using usernames and passwords, with role-based access control for different user types (admin, user, etc.).

3.4 System Requirements for Different User Roles

- **Admin User**: Needs full access to the system to manage appointments, users, and settings.
- **Regular User**: Can schedule, update, or cancel appointments but cannot modify system settings.
- **Guest User**: If the system has a guest access option, they may be limited to booking appointments but without login privileges.

PHP (Hypertext Preprocessor)

PHP is a widely-used open-source general-purpose scripting language that is especially suited for web development and can be embedded into HTML. It is often used to develop dynamic and interactive websites. PHP scripts are executed on the server, meaning the code is processed on the server before

sending HTML back to the client. PHP is compatible with almost all servers used today (e.g., Apache, IIS, etc.), and it can connect to different database systems such as MySQL, MariaDB, PostgreSQL, and more. PHP syntax is similar to other programming languages like Perl and C. Its flexibility and simplicity make it one of the most popular web development tools.

PhpMyAdmin

PhpMyAdmin is a free, open-source tool written in PHP, used to handle the administration of MySQL or MariaDB with the use of a web browser. It supports a wide range of operations on MySQL and MariaDB, including database creation, modification, and deletion. It allows for easy management of tables, indexes, privileges, and other aspects of databases, through a simple web-based interface. Advanced operations, such as SQL queries and database backup/export, can also be performed. It is particularly helpful for web developers or administrators who prefer a graphical user interface over manually working with the MySQL command line.

HTML (HyperText Markup Language)

HTML is the standard markup language for creating web pages and web applications. It describes the structure of web pages using elements represented by tags. HTML is the backbone of all web content, providing the basic building blocks of a web page, such as headings, paragraphs, images, links, forms, etc. HTML elements form the web page's structure, and combined with CSS (Cascading Style Sheets) and JavaScript, it creates rich and interactive web pages. HTML5 is the latest version, offering new features like multimedia handling, APIs, and improved accessibility.

CSS (Cascading Style Sheets)

CSS is a stylesheet language used to control the presentation of HTML elements, including layout, colors, fonts, spacing, and responsiveness. By separating the content from the design, CSS enhances web accessibility and reduces development complexity and repetition. It provides flexibility and power in adjusting how the web content looks on different devices (desktops, tablets, phones, etc.) and enables styles to be shared across multiple pages, making website maintenance easier. CSS frameworks such as Bootstrap offer pre-designed components and layouts, making it faster to build and design responsive web pages.

JavaScript

JavaScript is a high-level, versatile scripting language primarily used to create and control dynamic website content. Along with HTML and CSS, it forms the core technologies for building web pages and applications. Unlike HTML and CSS, which are markup and styling languages, JavaScript enables developers to make web pages interactive. This includes responding to user inputs, modifying content on the fly, animating elements, validating forms, creating games, and building complex web applications.

SYSTEM DESIGN

4.1 Data Design and Modelling

System design is the process of defining the architecture, components, modules, interface and data for a system to satisfy specified requirements. It a solution to an approach compared to system analysis which is it translates these "what is" orientation. System requirements into way of making them operational. The design phase focuses on detailed implementation of the system recommended in the feasibility study. Planning of system or to replace or complement an existing system. But before this, planning should be done. It must be thoroughly understood about the old system and determine how computers can make its operations more effective. The importance of system design is the quality. Design is the place where quality is fostered in the software development. Design representation of software provides us with that can be assessed for quality. System design is a transaction from a user- oriented documents to a programmer or database personal. It is a creative activity in both art and technology. It involves the following procedures, they are

• Database Design

• Input Design

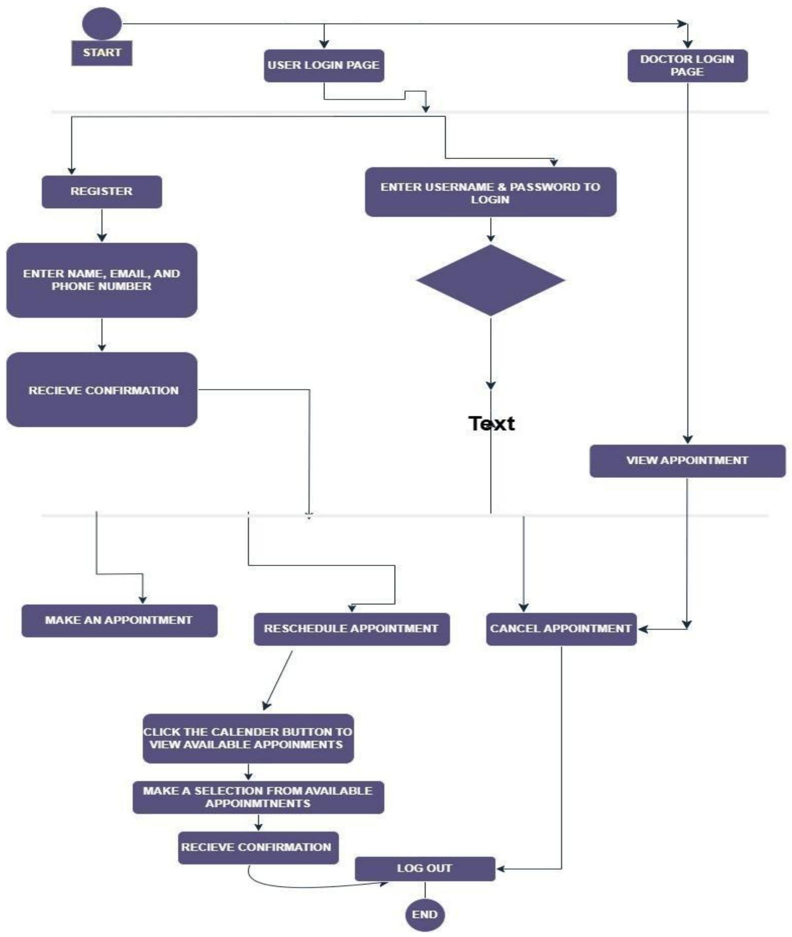
• Output Design

4.1.1 The Data Flow Diagram

The Data Flow Diagram (DFD) for an appointment scheduler system illustrates how data flows within the system, starting from high-level external interactions down to specific processes. In the Level 0 DFD, the Context Diagram highlights the interactions between external entities—the User and Service Provider—and the Appointment Scheduler System. Here, users can request actions such as booking, cancelling, or rescheduling appointments, while service providers manage their availability and appointments. The system returns confirmation messages, appointment details, available slots, and reminders to both users and providers. In Level 1, the DFD further breaks down the system into key processes: Manage User Requests, Manage Provider Availability, Notification System, Conflict Checker, and Time Zone Conversion. These processes interact with various data stores—such as the User Data Store, Provider Data Store, Appointment Data Store, and Notification Data Store—to handle specific functions. For instance, the Manage User Requests process handles all booking and rescheduling actions. When a user submits a booking request, the system checks availability and verifies that there are no conflicts. The Manage Provider Availability process allows providers to update or set available slots, which the system then shows to users searching for appointments. The Notification System sends appointment reminders and confirmations to ensure users and providers receive timely updates, while the Conflict Checker prevents overlapping appointments. The Time Zone Conversion process adjusts appointment times for each user's and provider's local time zone, making scheduling efficient across regions. At Level 2, the DFD explores more detailed actions within these processes. For example, the Manage User Requests process is broken down

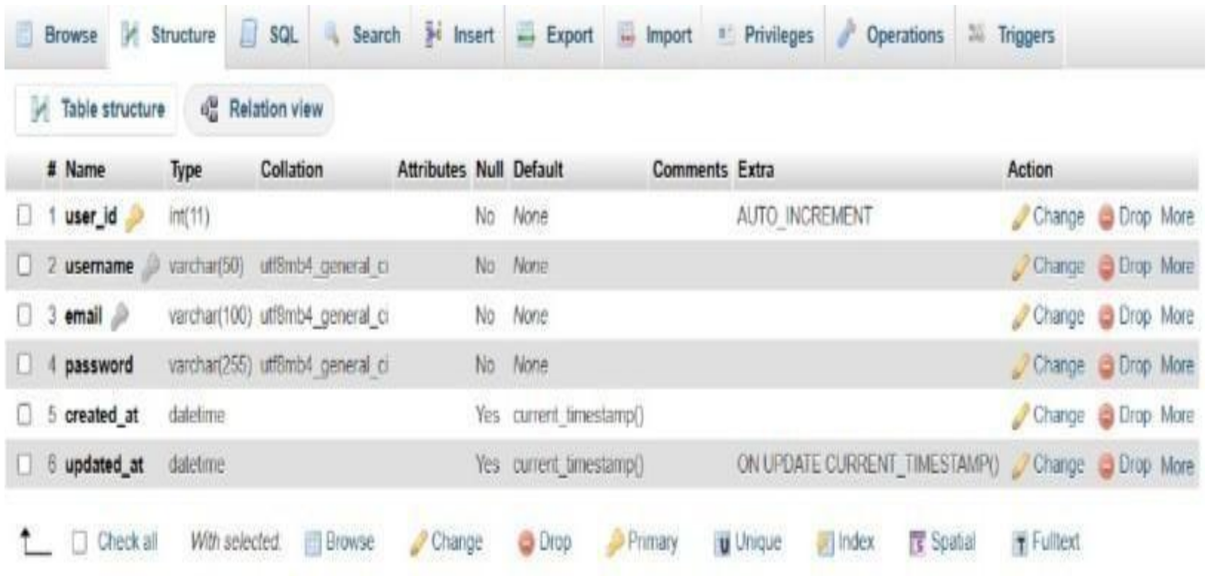
into smaller steps: Search Availability, Book Appointment, and Cancel/Reschedule Appointment. When a user searches for available slots, the system queries the Provider Data Store and returns open slots based on search criteria. For booking, once the user selects a time slot, the Conflict Checker ensures no overlapping bookings, stores the appointment if no conflicts exist, and triggers a notification to confirm the booking. Similarly, cancellation or rescheduling actions update the Appointment Data Store and notify all involved parties.

Overall, the DFD for this appointment scheduler system provides a clear representation of how different components and processes interact to handle user requests, manage availability, prevent conflicts, and communicate through notifications. It allows for efficient, scalable scheduling while ensuring data integrity and consistency across services.



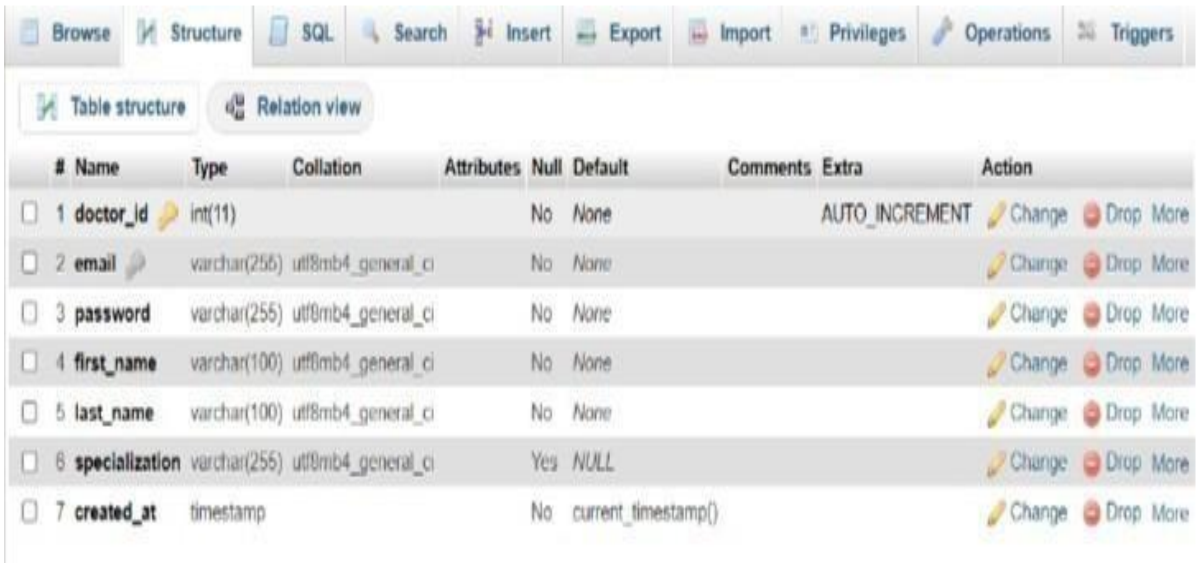
4.1.2 Schema Design

USER TABLE



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	user_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	username	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
3	email	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
4	password	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
5	created_at	datetime			Yes	current_timestamp()			Change Drop More
6	updated_at	datetime			Yes	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()	Change Drop More

APPOINTMENTS TABLE



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	doctor_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	email	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
3	password	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
4	first_name	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
5	last_name	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
6	specialization	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
7	created_at	timestamp			No	current_timestamp()			Change Drop More

4.2 Input Design

Input design is one of the most expensive phases of the operation of computerized system and often the major problem of a usually. A larger number of problems with a system can be traced back to fault input design and methods. Needless to say, therefore that output data is the block of a system and has to be analysed and designed consideration. It is the process of converting the user-oriented description of into a computer-based business information system inputs of input design is to create to a programmer-oriented specification. The objective errors. An input layout that is easy to follow and prevent operator. It covers all phases of input from creation of initial data into actual entry of the data to the system for processing. The input design is the link that ties the system into world of its users. The user interface design is very important for any application. The interface design defines how the software communication within itself, to system that interpreted with it and with human who use it. The goal of designing input data is to make the automation as easy and free from errors as possible. For providing a good input design for the application easy data input and selection features are adopted. The input design requirements such as user friendliness, also considered for the development of the project.

Requirements of Form Design:

- Identification and wording.
- Maximum readability and use
- Physical factors
- Order of data items.
- Easy of data entry
- Size and arrangement.
- Use of instructions

4.3 Output Design

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the user and to the other systems through outputs. In the output design it is determined how the information is to be displayed for immediate need and also the hard copy output. It is the most important and direct source information to the user. Thus, output design generally refers to the result and information that are generated by the system. For many end users output is the main reason for developing the system and the basis on which they are evaluate the usefulness of application. The objective of a system finds its shape in terms of the output. The analysis of the objective of the system leads to determination of outputs. Outputs of a system can take various forms. The most common are reports, Screens, Printed form, Animations etc. The outputs also vary in terms of their contents, frequency, timing and format. The users of the output, its purpose and sequence of details to be are all considered. The output forms a system in the justification for its existence. If the outputs are inadequate in anyway, the system itself is inadequate. The basic requirements of output are that it should be accurate, timely and appropriate, in terms of content, medium and layout for its intended purpose. Hence it is necessary to design output so that the objectives of the system are met in the best possible manner. The outputs are in the form of reports When designing output, the system analyst must accomplish things like, to determine what information to be present, to decide whether to display or print the information and select the output medium to distribute the output to intended recipients. The output is the most important and direct source of information to the user. So it should be provided in a most efficient formatted way. An efficient and intelligent output of the system improves the relationship between the user and the system and help in decision making.

4.4 Basic Modules2

- **User Management Module**

Purpose: Manages user accounts, user profiles, authentication, and authorization. Allows users to register, log in, edit profiles, and manage their personal details securely.

- **Service Provider Management Module**

Purpose: Manages the creation and maintenance of service provider profiles, including availability schedules, specialties, and other relevant details.

- **Scheduling Module**

Purpose: Handles the booking, cancellation, and rescheduling of appointments. This module ensures smooth management of appointment slots for users and service providers.

- **Notification Module**

Purpose: Sends notifications, including appointment confirmations, reminders, and updates via email, SMS, or push notifications.

- **Conflict Resolution Module**

Purpose: Checks for scheduling conflicts and prevents double bookings by validating appointment slots before confirmations are made.

- **Time Zone and Localization Module**

Purpose: Manages time zone conversions and adjusts appointment times for users and service providers based on their respective locations, ensuring accurate scheduling.

- **Reporting and Analytics Module**

Purpose: Provides reports and analytics for appointment trends, user activity, service provider performance, and other key metrics to offer insights for optimization.

IMPLEMENTATION AND TESTING

5.1 Implementation Approaches

The implementation of an appointment scheduler system requires a modular approach, focusing on setting up the environment, building individual modules, and integrating them into a cohesive platform. First, the tech stack is chosen, and a backend environment is configured with relational and in-memory databases for efficient data management and caching. Core modules, such as user management, scheduling, provider availability, and notification services, are built independently, with RESTful or GraphQL APIs facilitating communication between them. The scheduling module incorporates conflict-checking logic, while the notification module integrates with third-party services to send confirmations and reminders. The system also includes time zone management to ensure accurate scheduling across regions. Once all modules are built, they are integrated and tested, with caching and optimization steps ensuring quick access to data. A responsive frontend interface provides users with a seamless experience for booking, cancelling, and rescheduling appointments. Finally, CI/CD pipelines are established for automated testing and deployment, and monitoring tools are used to track system performance and maintain high availability.

- **Stage Implementation**

A staged implementation approach for an appointment scheduler system allows for incremental development, testing, and deployment, ensuring each phase functions correctly before moving to the next. Initially, the project starts with Stage 1: Core Setup and User Management, where the tech stack, databases, and core environment are configured, along with implementing basic user registration, login, and authentication features. In Stage 2: Provider and Availability Management, the focus shifts to allowing service providers to set and update availability, while users gain access to view open slots. Stage 3: Scheduling and Conflict Resolution introduces appointment booking, cancelling, and rescheduling features, with conflict-checking logic to prevent double bookings. Stage 4: Notifications and Time Zone Management adds essential notification features, integrating with third-party services to send confirmations and reminders and implementing time zone conversions to ensure accuracy across regions. Stage 5: Frontend and User Interface involves building a responsive interface for seamless booking and account management. Finally, Stage 6: Testing, Optimization, and Deployment ensures system reliability through testing (unit, integration, performance) and monitoring setup before live deployment, supported by continuous integration and deployment pipelines to maintain system stability and scalability. This phased approach enables a steady progression with opportunities for adjustments and optimizations along the way.

rejected depending on how it gathers confidence among the users. The implementation stage involves the following tasks.

6.1.1 Implementation methods

Implementing an appointment scheduler involves creating a system where users can seamlessly schedule, reschedule, or cancel appointments. First, it’s essential to outline key features, such as user authentication to ensure secure login and profile management, appointment management for creating and adjusting bookings, and availability management to display open time slots. Notifications, via email or SMS, help remind users of their appointments, reducing no-shows. Integrating with popular calendar systems like Google Calendar can enhance the user experience by allowing for easy sync. An admin panel enables administrators to oversee user management, appointment details, and scheduling adjustments. The technology stack can involve HTML, CSS, and JavaScript for the front end, with backend support from frameworks like Node.js, Django, or Laravel, and a database such as MySQL or MongoDB to manage data. Together, these elements create a responsive, user-friendly appointment scheduler that meets both user and administrative needs efficiently.

6.1.2 Implementation plan

The Implementation Plan describes how the information system will be deployed, installed and transitioned into an operational system. The plan contains an overview of the system, a brief description of the major tasks involved in the implementation, the overall resources needed to support the implementation effort, and any site-specific implementation requirements. The plan is developed during the Design Phase and is updated during the Development Phase the final version is provided in the Integration and Test Phase and is used for guidance during the implementation phase.

6.2 Testing Methods

Testing is the process of examining the software to compare the actual behaviour with that of the excepted behaviour. The major goal of software testing is to demonstrate that faults are not present. In order to achieve this goal, the tester executes the program with the intent of finding errors. Though testing cannot show absence of errors but by not showing their presence it is considered that these are not present. System testing is the first Stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operations commences. Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct and the goal will be successfully achieved. A series of testing are performed for the proposed system before the proposed system is ready for user acceptance testing.

Software testing is an integral part of to ensure software quality, some software organizations are reluctant to include testing in their software cycle, because they are afraid of the high cost associated with the software testing. There are several factors that attribute the cost of software testing. Creating and maintaining large number of test cases is a time- consuming process.

Once the source code has been generated, the program should be executed before the customer gets it with the specific intend of fining and removing all errors, test must be designed using disciplined techniques. Testing techniques provides the systematic guidance for designing to uncover the errors in the program behaviour function and performance the following steps to be done:

- Execute the integral logic of the software components

- Execute the input and output domains of the program to uncover errors

Software reliability is defined as the probability that the software will not undergo failure for a specified time under specified condition. Failure is the inability of a system or a component to perform a required function according to its specification. Different levels of testing were employed for software to make an error free, fault free and reliable. Basically, in software testing four type of testing methods are adopted

Levels of testing

- Unit Testing
- Integration Testing
- Validations
- System Testing

Unit testing

In this each module is tested individually before integrating it to the final system. Unit test focuses verification in the smallest unit of software design in each module. This is also known as module testing as here each module is tested to check whether it is producing the desired output and to see if any error occurs. Unit testing is commonly automated, but may still be performed manually. The objective in unit testing is to isolate a unit and validate its correctness. A manual approach to unit testing may employ a step-by- step instructional document. However, automation is efficient for achieving this, and enables the many benefits listed in this article. Conversely, if not planned carefully, a careless manual unit test case may execute as an integration test case that involves many software components, and thus preclude the achievement of most if not all of the goals established for unit testing. Unit testing focuses verification efforts even in the smallest unit of software design in each module. This is known as “module testing”.

The modules of this project are tested separately. This testing is carried out in the programming style itself. In this testing each module is focused to work satisfactorily as regard to expected output from the module. There are some validation checks for the fields. Unit testing gives stress on the modules of the project independently of one another, to find errors. Different modules are tested against the specifications produced during the design of the modules. Unit testing is done to test the working of individual modules with test servers. Program unit is usually small enough that the programmer who developed it can test it in a great detail. Unit testing focuses first on that the modules to locate errors. These errors are verified and corrected and so that the unit perfectly fits to the project.

Integration testing

Integration testing (sometimes called integration and testing, abbreviated I and T) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the

integrated system ready for system testing. The purpose of integration testing is to verify functional, performance, and reliability requirements placed on major design items.

System testing

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black-box testing, and as such, should require no knowledge of the inner design of the code or logic. As a rule, system testing takes, as its input, all of the "integrated" software components that have passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together or between any of the assemblages and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the "inter- assemblages" and also within the system as a whole. System testing focuses on testing the system as a whole. System Testing is a crucial step in Quality Management Process. In the Software Development Life Cycle, System Testing is the first level where the System is tested as a whole. The System is tested to verify whether it meets the functional and technical requirements. The application/System is tested in an environment that closely resembles the production environment where the application will be finally deployed. The perquisites for System Testing are:

- All the components should have been successfully Unit Tested.
- All the components should have been successfully integrated.

User acceptance testing

The system was tested by a small client community to see if the program met the requirements defined the analysis stage. It was fond to be satisfactory. In this phase, the system is fully tested by the client community against the requirements defined in the analysis and design stages, corrections are made as required, and the production system is built. User acceptance of the system is key factor for success of the system. User acceptance of a system is a key factor to success of any system. The system under consideration was tested for user acceptance by constantly keeping in touch with the prospective system user at the time of developing and making changes whenever required. This is done with regard to the following points.

- Input screen design.
- Output screen design.
- Format of the report and other

6.3 **Coding Details**

Index.php

```
<ophp
session_start();
```

```
// Check if the user is logged in, if not redirect to login page
if (!isset($_SESSION['logged_in']) || $_SESSION
    ['logged_in'] !== true) { header("Location: login.php");
    exit();
}

// Optional: Retrieve user name from session or database
$user_name = isset($_SESSION['user_name']) o $_SESSION
['user_name'] : "User"; o>

<!DOCTYflE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Appointment Scheduler</title>
    <link rel="stylesheet" href="index.css">
    <style>
        /* Add gradient background to the entire
        page */ body {
            margin: 0;
            padding: 0;
            font-family: Arial, sans-serif;
            background: linear-gradient(90deg,
                                                        rgba(112,109,158,0.765143557422969
2) 0%, rgba(0,0,0,0.7175245098039216) 34%,
rgba(255,170,170,0.6755077030812324) 100%);
            color: #fff;
            text-align: center;
        }

        /* Center the logo */
        .logo-link {
            display: block;
            margin: 20px auto;
        }

        .top-left {
            max-width: 100px;
            height: auto;
        }

        .container {
            background: rgba(255, 255, 255, 0.8);
            display: inline-block;
            padding: 20px;
            border-radius:
            10px;
            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
            color: #000;
```

```
}

header h1 {
     margin: 0;
     padding: 10px 0;
}



form {
     display: flex;
     flex-direction:
     column; align-
     items: center;
}

.form-group {
     margin-bottom: 15px;
     width: 100%;
     max-width: 400px;
}

label {
     display: block;
     margin-bottom:
     5px; font-weight:
     bold;
}

input[type="text"],
 input[type="email
            "],
 input[type="date"
            ],
 input[type="time"
 ] { width: 100%;
     padding: 10px;
     border: 1px solid #
     ddd; border-radius:
     5px;
}

button {
     background-color: #2575fc;
     color: white;
     border: none;
     padding: 10px
     20px; border-
     radius: 5px;
     cursor: pointer;
```

```
                font-size: 16px;
            }

            button:hover {
                background-color: #6a11cb;
            }
        </style>
    </head>
    <body>
        <a href="/" class="logo-link">
            <img
                      src="https://images.shiksha.com/mediadata/images/1607938223phpFW2T
    mn.jpeg" alt="Website Logo" class="top-left">
        </a>
        <div class="container">
            <header>
                <h1>Welcome, <ophp     echo
                                  htmlspecialchars($user_name);      o>!  Schedule
                                  Your Appointment</h1>
            </header>
            <main>
                <!-- Form for booking an appointment -->
                <form action="submit_appointment.php" method="post">
                    <div class="form-group">
                        <label for="name">Full Name</label>
                        <input type="text" id="name"name="name" placeholder="John
                                Doe" value="<ophp echo htmlspecialchars($user_name);
    o>" required>
                    </div>
                    <div class="form-group">
                        <label for="email">Email Address</label>


                        <input          type="email"        id="email"
                                        name="email"
    placeholder="john.doe@example.com" required>
                    </div>
                    <div class="form-group">
                        <label for="date">Select Date</label>
                        <input type="date" id="date" name="date" required>
                    </div>
                    <div class="form-group">
                        <label for="time">Select Time</label>
                        <input type="time" id="time" name="time" required>
                    </div>
                    <div class="form-group">
                        <button type="submit">Book Appointment</button>
                    </div>
                </form>

                <!-- Form for canceling an appointment -->
```

```
<div  class="form-group">
     <a href="view_appointments.php">
          <button>View Appointments</button>
     </a>
</div>
          </main>
     </div>
</body>
</html>
```

CANCELLING.PHP

```
<ophp
session_start();

// Database
connection try {
     $pdo  =  new
flDO('mysql:host=localhost;port=3307;dbname=appointment_scheduler', 'root', '');
     $pdo->setAttribute(flDO::ATTR_ERRMODE,       flDO::ERRMODE_EXCEflTION);
} catch (flDOException $e) {
     die("Database  connection  failed:  " . $e->getMessage());
}

// Check if the form was submitted
 if ($_SERVER["REQUEST_METHOD"] == "flOST") {
   if (isset($_flOST['id'])) {
          $appointment_id    =      htmlspecialchars($_flOST['id']);

          // Delete the appointment from the database
          $sql = "DELETE FROM appointments WHERE id = o";
          $stmt = $pdo->prepare($sql);

          if ($stmt->execute([$appointment_id])) {
               // Redirect back to the appointments page after cancellation
               header("Location:      view_appointments.phpomessage=Appointment
                                      canceled successfully.");
               exit();
          } else {
               echo "Error canceling appointment. fllease try again.";
          }
     }
} else {
     header("Location:
     view_appointments.php"); exit();
}
o>
```

SUBMIT APPOINTMENT.PHP

```
<ophp
// submit_appointment.php

use flʜflMailer\flʜflMailer\flʜ
flMailer; use flʜflMailer\flʜ
flMailer\Exception;

// Include flʜflMailer files (adjust the path if necessary)
require 'vendor/autoload.php';

// Get the booking details from the query parameters
$email = $_GET['email'];
$name = $_GET['name'];
$doctor_email = $_GET['doctor_email'];
$date = $_GET['date'];
$time = $_GET['time'];

// Create a new flʜflMailer instance
$mail = new flʜflMailer(true);

try {
    //Server settings
    $mail->isSMTfl();                                    // Set mailer to use SMTfl
    $mail->ʜost = 'smtp.gmail.com';                           // Set the SMTfl
server (replace with your SMTfl server)
    $mail->SMTflAuth = true;                             // Enable SMTfl authentication
    $mail->Username = 'acheduler19456@gmail.com';              // SMTfl username
    $mail->flassword = 'Abbevi@19456';                      // SMTfl password
    $mail->SMTflSecure = flʜflMailer::EɴCRYflTIOɴ_STARTTLS; // Enable TLS encryption
    $mail->flort = 587;                                    // TCfl port to connect to

    //Recipients
    $mail->setFrom('no-reply@example.com', 'Appointment Booking'); // From address
    $mail->addAddress($email, $name);                      // Add recipient (patient)

    // Content
    $mail->isʜTML(true);                                 // Set email format to ʜTML
    $mail->Subject = 'Appointment Booking Confirmation';
    $mail->Body = "
        <h3>Your appointment has been successfully booked!</h3>
        <p><strong>Doctor:</strong> $doctor_email</p>
        <p><strong>Date:</strong> $date</p>
        <p><strong>Time:</strong> $time</p>
        <p>Thank you for booking with us. We look forward to seeing you!</p>
    ";

    // Send the email
    $mail->send();
    echo 'Confirmation email has been sent.';
} catch (Exception $e) {
    echo "Message could not be sent. Mailer Error: {$mail->ErrorInfo}";
}
```

o>

BOOKING.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Book Your Appointment</title>
  <link rel="stylesheet" href="style.css">
  <link  rel="stylesheet"  href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css">
  <style
    >
    body
    {
      font-family: 'Arial', sans-serif;
      margin: 0;
      padding: 0;
      box-sizing: border-
      box; display: flex;
      flex-direction:
      column; min-height:
      100vh;
    }
    header {
      background-color:
      #333; color: #fff;
      padding: 10px
      20px; position:
      fixed; width:
      100%;
      z-index: 100;
      transition: background-color 0.3s;
    }
    header.scrolled {
      background-color: rgba(51, 51, 51, 0.9);
    }
    .navbar {
      display: flex;
      justify-content: space-
      between; align-items: center;
    }
    .nav-links {
      list-style: none;
      display: flex;
    }
```

```
.nav-links li {
 margin-left:
 20px;
}
.nav-links a {
 color: #fff;
 text-decoration: none;
 transition: color 0.3s;
}
.nav-links a:hover {
 color: #4CAF50;
}
.booking-section {
 margin-top: 80px; /* Space for fixed header */
 padding: 40px;
 text-align: center;
 flex: 1;
 background: linear-gradient(to bottom, #f9f9f9,
 #eaeaea); box-shadow: 0 0 20px rgba(0, 0, 0, 0.1);


}
h2 {
 margin-bottom: 20px;
 color: #333;
}
.booking-form {
 max-width:
 600px; margin: 0
 auto; display:
 flex;
 flex-direction: column;
 gap: 15px;
}
.booking-form select, .booking-form button {
 padding: 10px;
 font-size: 16px;
 border: 1px solid
 #ccc; border-radius:
 4px; transition:
 border 0.3s;
}
.booking-form
 select:focus   {   border:
 1px    solid    #4CAF50;
 outline: none;
}
.booking-form button {
 background-color:
 #4CAF50; color: white;
```

```
      border: none;
      cursor: pointer;
      transition: background-color 0.3s;
    }
    .booking-form
      button:hover {
      background-color:
      #45a049;
    }
    /* Footer styles */
    footer {
      background-color:
      #333; color: #fff;
      padding: 20px;
      text-align:
      center; position:
      relative;
    }
    .footer-content
      { display:
      flex;
      justify-content: space-
      between; flex-wrap: wrap;
      align-items: center;
    }
    .footer-links a {
      color: #fff;
      margin: 0 10px;
      text-decoration: none;
      transition: color 0.3s;
    }
    .footer-links a:hover {

      color: #4CAF50;
    }
    .social-media a {
      margin: 0 10px;
      color: #fff;
      transition: color
      0.3s;
    }
    .social-media a:hover {
      color: #4CAF50;
    }
  </style>
</head>
<body>

  <!-- Header -->
```

```
 <header id="header">
  <nav class="navbar">
    <h1>
   <div class="logo">
     <img                                                            src="https://cdn.shopify.com/app-
store/listing_images/35b63cb98175322a555e035845626664/icon/CP7xiPDSyP8CEAE=.png"
                                                                                                    alt="Log
o" style="width: 40px; height: 40px; vertical-align: middle; margin-right: 10px;">
     ABBEVI Bookings
   </div>
  </h1>

   <ul class="nav-links">
     <li><a href="index.html">Home</a></li>
     <li><a href="#services">Services</a></li>
     <li><a href="#pricing">Pricing</a></li>
     <li><a href="#testimonials">Testimonials</a></li>
     <li><a href="#contact">Contact Us</a></li>
     <li><a href="login.html">Login</a></li>
     <li><a href="register.html" class="signup">Sign Up</a></li>
   </ul>
  </nav>
 </header>

 <!-- Booking Section -->
 <section class="booking-section" id="booking">
  <h2>Book Your Appointment</h2>
  <form class="booking-form" action="#" method="POST">
   <label for="service">Select Service:</label>
   <select id="service" name="service" required onchange="redirectToService()">
     <option value="">--Choose a Service--</option>
     <option value="medical.html">Medical Appointments</option>
     <option value="event.html">Event Bookings</option>
   </select>
  </form>
 </section>

 <!-- Footer -->
 <footer>
  <div class="footer-content">


   <div class="social-media">
     <a href="#"><i class="fab fa-linkedin"></i></a>
     <a href="#"><i class="fab fa-twitter"></i></a>
     <a href="#"><i class="fab fa-facebook"></i></a>
   </div>
   <div class="footer-links">
     <a href="#">About Us</a>
     <a href="#">Privacy Policy</a>
```

```
        <a href="#">Terms of Service</a>
        <a href="#">Contact</a>
      </div>
    </div>
  </footer>

  <script>
    // Change header background on scroll
    window.onscroll = function() {
      const header = document.getElementById('header');
      if (document.body.scrollTop > 50 || document.documentElement.scrollTop
        > 50) { header.classList.add('scrolled');
      } else {
        header.classList.remove('scrolled'
        );
      }
    };

    // Redirect to the selected service page
    function redirectToService() {
      const serviceSelect =
      document.getElementById('service'); const
      selectedService = serviceSelect.value;
      if (selectedService) {
        window.location.href = selectedService; // Redirect to the selected page
      }
    }
  </script>

</body>
</html>
```

REGISTER.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Sign Up - ABBEVI Bookings</title>
    <link rel="stylesheet" href="stylereg.css">
    <link          rel="stylesheet"          href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-
beta3/css/all.min.css">
  <style>
    body
    {
     font-family: 'Arial', sans-serif;
     margin: 0;
     padding: 0;
     box-sizing: border-box;
     display: flex;
     flex-direction: column;
     min-height: 100vh;
    }
     header {
     background-color: #333;
     color: #fff;
     padding: 10px 20px;
     position: fixed;
     width: 100%;
     z-index: 100;
     transition: background-color 0.3s;
    }
    header.scrolled {
     background-color: rgba(51, 51, 51, 0.9);
    }
    .navbar {      display:
     flex; justify-c
ontent: space-between;
     align-items: center;
    }
    .nav-links {
     list-style: none;
     display: flex;
    }
    .nav-links li {
     margin-left: 20px;
    }
    .nav-links a {
     color: #fff;
     text-decoration: none;
     transition: color 0.3s;
    }
    .nav-links a:hover {
     color: #4CAF50;
    }

    .content-section {
     margin-top: 80px;
     padding: 40px;
     flex: 1;
```

```
      background: linear-gradient(to bottom, #f9f9f9, #eaeaea);
    }
    h2 {
      margin-bottom: 20px;

      color: #333;
    }
    .appointment-card {
      border: 1px solid #ccc;
      border-radius: 4px;
      background-color: #fff;
      box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
      cursor: pointer;
      transition: transform 0.3s;
    }
    .appointment-card:hover {
      transform: scale(1.05);
    }
    .appointment-card img {
      border-radius: 50%;
      width: 80px;
      height: 80px;
      margin: 15px auto;
    }
    .appointment-card h3 {
      color: #333;
      margin-bottom: 10px;
    }
    .appointment-card p {
      margin: 0;
      color: #555;
    }
        .social-media a {
      margin: 0 10px;
      color: #fff;
      transition: color 0.3s;
    }
    .social-media a:hover {
      color: #4CAF50;
    }
  </style></head>
<body>

    <!-- Header -->
    <header>
        <nav class="navbar">
            <h1>
        <div class="logo">
          <img src="https://cdn.shopify.com/app-store/listing_images/35b63cb98175322a555e035845626664/icon/CP7xiPDSyP8CEAE=.png" alt="Logo" style="width: 40px; height: 40px; vertical-align: middle; margin-right: 10px;">
          ABBEVI Bookings
        </div>
      </h1>


          <ul class="nav-links">
            <li><a href="index.html">Home</a></li>
            <li><a href="#services">Services</a></li>
            <li><a href="#pricing">Pricing</a></li>
            <li><a href="#testimonials">Testimonials</a></li>
            <li><a href="#contact">Contact Us</a></li>
            <li><a href="login.html">Login</a></li>
            <li><a href="#" class="signup">Sign Up</a></li>
          </ul>
        </nav>
      </header>

    <!-- Signup Section -->
    <section class="signup-section">
      <div class="container">
        <h1>Create Your Account</h1>
        <form action="#" method="POST">
          <div class="form-group">
            <label for="name">Full Name</label>
            <input type="text" id="name" name="name" required placeholder="Enter your full name">
          </div>
          <div class="form-group">
            <label for="email">Email Address</label>
            <input type="email" id="email" name="email" required placeholder="Enter your email address">
          </div>
          <div class="form-group">
            <label for="password">Password</label>
            <input type="password" id="password" name="password" required placeholder="Create a password">
          </div>
          <div class="form-group">
            <label for="confirm-password">Confirm Password</label>
            <input type="password" id="confirm-password" name="confirm-password" required placeholder="Confirm your password">
          </div>
          <button type="submit" class="btn-primary">Sign Up</button>
        </form>
        <p class="terms">By signing up, you agree to our <a href="#">Terms of Service</a> and <a
href="#">Privacy Policy</a>.</p>
        <p>Already have an account? <a href="login.html">Login here</a>.</p>
      </div>
    </section>

    <!-- Footer -->
  <footer>
    <div class="footer-content">
      <div class="social-media">

        <a href="#"><i class="fab fa-linkedin"></i></a>
        <a href="#"><i class="fab fa-twitter"></i></a>
        <a href="#"><i class="fab fa-facebook"></i></a>
      </div>
      <div class="footer-links">
        <a href="#">Privacy Policy</a>
        <a href="#">Terms of Service</a>
        <a href="#">Contact</a>
      </div>
    </div>
  </footer>


</body>
</html>
```

LOGIN.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Login - Appointment Scheduler</title>
  <link rel="stylesheet" href="login.css">
  <link  rel="stylesheet"  href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css">
</head>
<body>

  <!-- Header -->
  <header>
    <nav class="navbar">
<h1>
      <div class="logo">
```

```
        <img                                                                   src="https://cdn.shopify.com/app-
store/listing_images/35b63cb98175322a555e035845626664/icon/CP7xiPDSyP8CEAE=.png"
                                                                                                    alt="Log
o" style="width: 40px; height: 40px; vertical-align: middle; margin-right: 10px;">
        ABBEVI Bookings
      </div>
    </h1>

    <ul class="nav-links">
      <li><a href="index.html">Home</a></li>
      <li><a href="#">Services</a></li>
      <li><a href="#">Pricing</a></li>
      <li><a href="#">Contact Us</a></li>
    </ul>
  </nav>
</header>

<!-- Login Section -->


<section class="login-section">
  <div class="login-container">
    <div class="login-box">
      <img    src="https://img.freepik.com/premium-vector/user-profile-login-access-authentication-
icon_690577- 203.jpg" alt="Logo" class="logo-img">
      <h2>Sign in to ABBEVI Bookings</h2>

      <form action="#" method="POST">
        <div class="input-group">
          <i class="fas fa-envelope"></i>
          <input type="email" id="email" name="email" required placeholder="Email or phone">
        </div>

        <div class="input-group">
          <i class="fas fa-lock"></i>
          <input type="password" id="password" name="password" required placeholder="Enter your password">
        </div>

        <div class="remember-me">
          <input type="checkbox" id="remember-me" name="remember-me">
          <label for="remember-me">Remember Me</label>
        </div>

        <div class="forgot-password">
          <a href="#">Forgot Password?</a>
        </div>

        <button type="submit" class="btn-primary">Login</button>
        <p class="signup-link">Don't have an account? <a href="register.html">Sign up here</a></p>
      </form>
    </div>
  </div>
</section>

<!-- Footer -->
<footer>
  <div class="footer-content">
    <div class="social-media">
      <a href="#"><i class="fab fa-linkedin"></i></a>
      <a href="#"><i class="fab fa-twitter"></i></a>
      <a href="#"><i class="fab fa-facebook"></i></a>
    </div>
    <div class="footer-links">
      <a href="#">Privacy Policy</a>
      <a href="#">Terms of Service</a>
      <a href="#">Contact</a>
    </div>
  </div>
</footer>

</body>
</html>
```

MEDICAL.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Medical Appointment Information</title>
  <link rel="stylesheet" href="style3.css">
  <link            rel="stylesheet"            href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-
beta3/css/all.min.css">
  <link                                                                        rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
  <style>
    body
    {
      font-family: 'Arial', sans-serif;
      margin: 0;
      padding: 0;
      box-sizing: border-box;
```

```
   display: flex;
   flex-direction: column;
   min-height: 100vh;
  }
  header {
   background-color: #333;
   color: #fff;
   padding: 10px 20px;
   position: fixed;
   width: 100%;
   z-index: 100;
   transition: background-color 0.3s;
  }
  header.scrolled {
   background-color: rgba(51, 51, 51, 0.9);
  }
  .navbar {     display:
   flex; justify-c
ontent: space-between;
   align-items: center;
  }
  .nav-links {
   list-style: none;
   display: flex;
  }
  .nav-links li {
   margin-left: 20px;
  }
  .nav-links a {

   color: #fff;
   text-decoration: none;
   transition: color 0.3s;
  }
  .nav-links a:hover {
   color: #4CAF50;
  }

  .content-section {
   margin-top: 80px; /* Space for fixed header */
   padding: 40px;
   flex: 1;
   background: linear-gradient(to bottom, #f9f9f9, #eaeaea);
  }
  h2 {
   margin-bottom: 20px;
   color: #333;
  }
  .doctor-card {
   border: 1px solid #ccc;
   border-radius: 4px;
   background-color: #fff;
   box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
   transition: transform 0.3s, box-shadow 0.3s;
   cursor: pointer; /* Makes it clear that the card is clickable */
  }
  .doctor-card:hover {
   transform: scale(1.05); /* Scale effect on hover */
   box-shadow: 0 4px 20px rgba(0, 0, 0, 0.2); /* Enhanced shadow on hover */
  }
  .doctor-card img {
   border-radius: 50%;
   width: 80px;
   height: 80px;
   margin: 15px auto;
  }
  footer {
   background-color: #333;
   color: #fff;
   padding: 20px;
   text-align: center;
   position: relative;
  }
  .footer-content {
   display: flex;
   justify-content: space-between;
   flex-wrap: wrap;
   align-items: center;
  }

  .footer-links a {
   color: #fff;
   margin: 0 10px;
   text-decoration: none;
   transition: color 0.3s;
  }
  .footer-links a:hover
   { color: #4CAF50;
  }
  .social-media a {
   margin: 0 10px;
   color: #fff;
```

```
      transition: color 0.3s;
    }
    .social-media a:hover {
      color: #4CAF50;
    }
  </style>
</head>
<body>

  <!-- Header -->
  <header id="header">
    <nav class="navbar">
        <h3>
      <div class="logo">
        <img src="https://cdn.shopify.com/app-store/listing_images/35b63cb98175322a555e035845626664/icon/CP7xiPDSyP8CEAE=.png" alt="Logo" style="width: 40px; height: 40px; vertical-align: middle; margin-right: 10px;">
        ABBEVI Bookings
      </div>
    </h3>
      <ul class="nav-links">
        <li><a href="index.html">Home</a></li>
        <li><a href="book.html">Book Appointment</a></li>
        <li><a href="#services">Services</a></li>
        <li><a href="#pricing">Pricing</a></li>
        <li><a href="#testimonials">Testimonials</a></li>
        <li><a href="#contact">Contact Us</a></li>
        <li><a href="login.html">Login</a></li>
        <li><a href="register.html" class="signup">Sign Up</a></li>
      </ul>
    </nav>
  </header>

  <!-- Content Section -->
  <section class="content-section" id="medical-info">
    <div class="container">
      <h2>Medical Appointment Information</h2>

      <div class="row">
        <div class="col-md-3 mb-4">
          <div class="doctor-card p-3 text-center" onclick="location.href='book.html'">
            <img src="https://www.hydatis.com/wp-content/uploads/2020/05/1.jpg" alt="Dr. Arjun Menon">
            <h3>Dr. Arjun Menon</h3>
            <p>Specialization: General Practitioner</p>
            <p>Clinic: Kerala Health Center</p>
            <p>Phone: +91 98765 43210</p>
            <p>Available Timings: Mon-Fri, 9 AM - 5 PM</p>
            <p>Email: arjun.menon@abbevi.com</p>
          </div>
        </div>
        <div class="col-md-3 mb-4">
          <div class="doctor-card p-3 text-center" onclick="location.href='book.html'">
            <img   src="https://www.capitaldigestivecare.com/wp-content/uploads/2022/10/John-Smith-5-x- 7-
scaled.jpg" alt="Dr. Sita Suresh">
            <h3>Dr. Sita Suresh</h3>
            <p>Specialization: Cardiology</p>
            <p>Clinic: Heart Care Hospital</p>
            <p>Phone: +91 99876 54321</p>
            <p>Available Timings: Tue, Thu, Sat, 10 AM - 3 PM</p>
            <p>Email: sita.suresh@abbevi.com</p>
          </div>
        </div>
        <div class="col-md-3 mb-4">
          <div class="doctor-card p-3 text-center" onclick="location.href='book.html'">
            <img              src="https://cdn.askapollo.com/live/images/doctors/cardiology/dr-balakrishnan-n-
cardiology-cochin.jpg" alt="Dr. Neeta Kumar">
            <h3>Dr. Neeta Kumar</h3>
            <p>Specialization: Dermatology</p>
            <p>Clinic: Skin Health Clinic</p>
            <p>Phone: +91 98760 12345</p>
            <p>Available Timings: Mon, Wed, Fri, 11 AM - 4 PM</p>
            <p>Email: neeta.kumar@abbevi.com</p>
          </div>
        </div>
        <div class="col-md-3 mb-4">
          <div class="doctor-card p-3 text-center" onclick="location.href='book.html'">
            <img                                                                src="https://cdn.prod.website-
files.com/659c9d2768dc328628d30423/65a2d3e9b93766fb64147dd3_Dr.%20KN%20Srinivasan.jpg"
alt="Dr. Rahul Pillai">
            <h3>Dr. Rahul Pillai</h3>
            <p>Specialization: Pediatrics</p>
            <p>Clinic: Child Care Center</p>
            <p>Phone: +91 91234 56789</p>
            <p>Available Timings: Mon-Fri, 8 AM - 1 PM</p>
            <p>Email: rahul.pillai@abbevi.com</p>
          </div>
        </div>

        <div class="col-md-3 mb-4">
          <div class="doctor-card p-3 text-center" onclick="location.href='book.html'">
            <img       src="https://cdn.askapollo.com/live/images/doctors/cardiology/dr-byomakesh-dikshit-
cardiology-in-bhubaneswar.png" alt="Dr. Leela Nair">
            <h3>Dr. Leela Nair</h3>
            <p>Specialization: Neurology</p>
            <p>Clinic: Neuro Health Clinic</p>
            <p>Phone: +91 99887 65432</p>
            <p>Available Timings: Mon, Wed, Fri, 10 AM - 2 PM</p>
            <p>Email: leela.nair@abbevi.com</p>
          </div>
        </div>
        <div class="col-md-3 mb-4">
          <div class="doctor-card p-3 text-center" onclick="location.href='book.html'">
            <img src="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSik8RqXeWGt7lRyx0jNyM81FflU8YU87GrNw&s" alt="Dr. Vinay Menon">
            <h3>Dr. Vinay Menon</h3>
            <p>Specialization: Orthopedics</p>
            <p>Clinic: Bone Care Center</p>
            <p>Phone: +91 96789 12345</p>
            <p>Available Timings: Tue, Thu, 9 AM - 4 PM</p>
            <p>Email: vinay.menon@abbevi.com</p>
          </div>
        </div>
        <div class="col-md-3 mb-4">
          <div class="doctor-card p-3 text-center" onclick="location.href='book.html'">
            <img src="https://png.pngtree.com/png-vector/20240112/ourlarge/pngtree-indian-lady-doctor- png-image_11436098.png" alt="Dr. Anita Roy">
            <h3>Dr. Anita Roy</h3>
            <p>Specialization: Gynecology</p>
            <p>Clinic: Women's Health Clinic</p>
            <p>Phone: +91 87654 32109</p>
            <p>Available Timings: Mon-Fri, 9 AM - 5 PM</p>
            <p>Email: anita.roy@abbevi.com</p>
          </div>
        </div>
        <div class="col-md-3 mb-4">
          <div class="doctor-card p-3 text-center" onclick="location.href='book.html'">
            <img src="https://www.citizenshospitals.com/static/uploads/130789a4-764e-4ee3-88fe-68f9278452d6-1692966652977.png" alt="Dr. Ajay Kumar">
            <h3>Dr. Ajay Kumar</h3>
            <p>Specialization: Psychiatry</p>
            <p>Clinic: Mind Wellness Center</p>
            <p>Phone: +91 91234 67890</p>
            <p>Available Timings: Mon-Fri, 10 AM - 6 PM</p>
            <p>Email: ajay.kumar@abbevi.com</p>
      </div>
    </div>
  </section>

  <!-- Footer -->
  <footer>
    <div class="footer-content">
      <div class="social-media">
        <a href="#"><i class="fab fa-linkedin"></i></a>
        <a href="#"><i class="fab fa-twitter"></i></a>
        <a href="#"><i class="fab fa-facebook"></i></a>
      </div>
      <div class="footer-links">
        <a href="#">About Us</a>
        <a href="#">Privacy Policy</a>
        <a href="#">Terms of Service</a>
        <a href="#">Contact</a>
      </div>
    </div>
  </footer>

  <script>
    // Change header background on scroll
    window.onscroll = function() {
      const header = document.getElementById('header');
      if (document.body.scrollTop > 50 || document.documentElement.scrollTop > 50) {
        header.classList.add('scrolled');
      } else {
        header.classList.remove('scrolled');
      }
    };
  </script>

</body>
</html>
```

STYLE.CSS

```
/* Global Styles */
body, html {
margin: 0;
  padding: 0;
  font-family: 'Arial', sans-
  serif; box-sizing: border-
  box; scroll-behavior:
  smooth;
}
```

```
header {
  background-color: #333;
  color: #fff;
  padding: 10px 20px;
  position: fixed;
  top: 0;
  width: 100%;
  z-index: 100;
  box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
}
.how-it-works
{
margin-left:3rem;
}

.navbar {
  display: flex;
  justify-content: space-between;
  align-items: center;
}

.nav-links {
 margin-left:115vh;
list-style: none;
display: flex;
position:absolute;
}

.nav-links li {
  margin-left: 20px;
}

.nav-links a {
  color: #fff;
  text-decoration: none;
  transition: color 0.3s ease;
}

.nav-links a:hover {
  color: #4CAF50;
}

.signup {
  background-color: #4CAF50;
  padding: 8px 16px;
  border-radius: 4px;
}
```

```
  background-color: #45a049;
}

/* Hero Section */
.hero {
  height: 100vh;
  background: url('hero-image.jpg') no-repeat center center/cover; display:
  flex;
  align-items: center;
  justify-content: center;
  color: #fff;
  text-align: center;
}

.hero-content h1 {
  font-size: 3rem;
  transition: all 0.5s ease-in-out;
}

.hero-content p {
  font-size: 1.2rem;
}

.cta-buttons .btn-primary, .cta-buttons .btn-secondary {
  padding: 10px 20px;
  border-radius: 4px;
  text-decoration: none;
  color: #fff;
  margin: 10px;
  transition: background-color 0.3s ease;
}
.btn-primary {
  background-color: #4CAF50;
}

.btn-primary:hover {
  background-color: #45a049;
}

.btn-secondary {
  background-color: #555;
}

.btn-secondary:hover {
  background-color: #333;
}
```

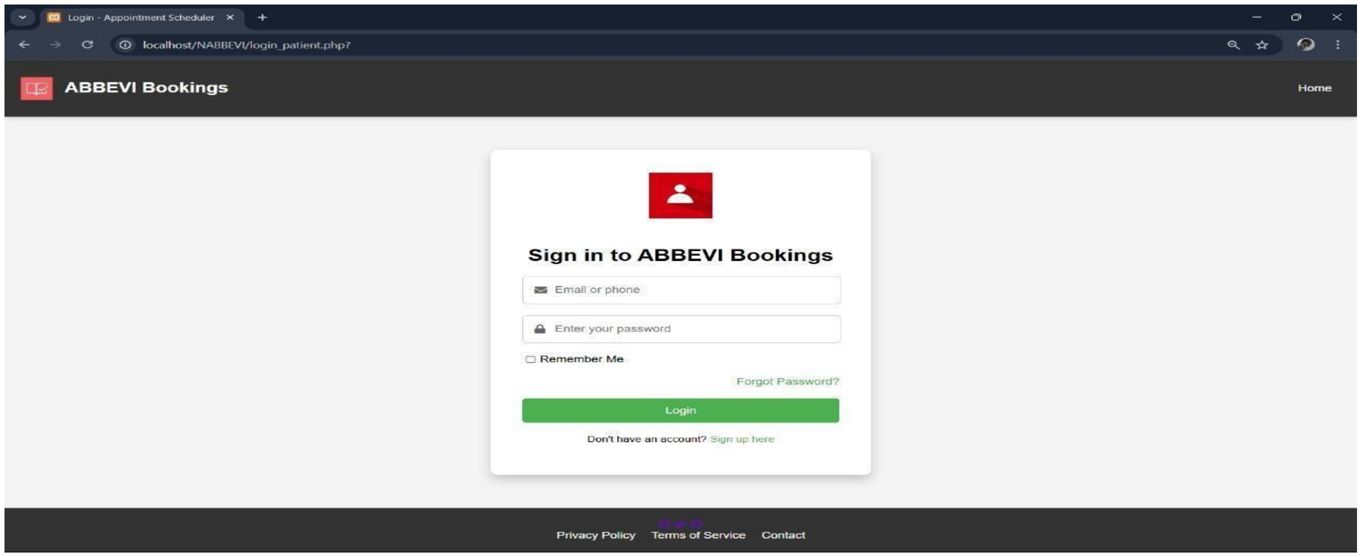
```
.features {
  padding: 50px 0;
  background-color: #f4f4f4;
  text-align: center;
}

.feature-cards {
  display: flex;
  justify-content: space-around;
  margin-top: 20px;
}

.feature-card {
  background-color: #fff;
  padding: 20px;
  border-radius: 8px;
  width: 30%;
  transition: transform 0.3s ease, background-color 0.3s ease;
}

.feature-card:hover {
  transform: scale(1.05);
  background-color: #e8f5e9;
}

/* Services Section */
.services {
  padding: 50px 0;
  background-color: #fff;
}
.service-cards {
  display: flex;
  justify-content: space-around;
}

.service-card {
  background-color: #f4f4f4;
  padding: 20px;
  border-radius: 8px;
  width: 22%;
  transition: background-color 0.3s ease;
}

.service-card:hover {
  background-color: #e0f7fa;
}

/* Pricing Section */
```

```
.pricing {
  padding: 50px 0;
  background-color: #f9fbe7;
}
.pricing-tables {
  display: flex;
  justify-content: space-around;
}

.pricing-table {
  background-color: #fff;
  padding: 20px;
  border-radius: 8px;
  width: 30%;
  transition: transform 0.3s ease, background-color 0.3s ease;
}

.pricing-table:hover {
  transform: scale(1.05);
  background-color: #f1f8e9;
}

.pricing-table h3 {
  margin-bottom: 10px;
}

.pricing-table   ul
  {        list-style:
  none;   padding:
  0;
}
.pricing-table ul li
  { margin: 5px 0;
}

/* Testimonials Section */
.testimonials {
  background-color: #fff;
  padding: 50px 0;
  text-align: center;
}

.testimonial-cards {
  display: flex;
  justify-content: center;
}

.testimonial-card {
  background-color: #f9f9f9;
  padding: 20px;
  border-radius: 8px;
  margin: 0 20px;
  width: 40%;
}
/* Newsletter Section */
.newsletter {
  background-color: #4CAF50;
  color: #fff;
  padding: 50px 0;
  text-align: center;
}

.newsletter input[type="email"] {
  padding: 10px;
  width: 300px;
  border: none;
  border-radius: 4px;
  margin-right: 10px;
}

.newsletter button {
  padding: 10px
  20px; border: none;
  border-radius: 4px;
  background-color: #fff;
  color: #4CAF50;
  cursor: pointer;
  transition: background-color 0.3s ease;
}

.newsletter button:hover {
  background-color: #f4f4f4;
}
/* Footer */
footer {
  background-color: #333;
  color: #fff;
  padding: 20px 0;
  text-align: center;
}

.footer-links {
  margin-top: 10px;
}
```

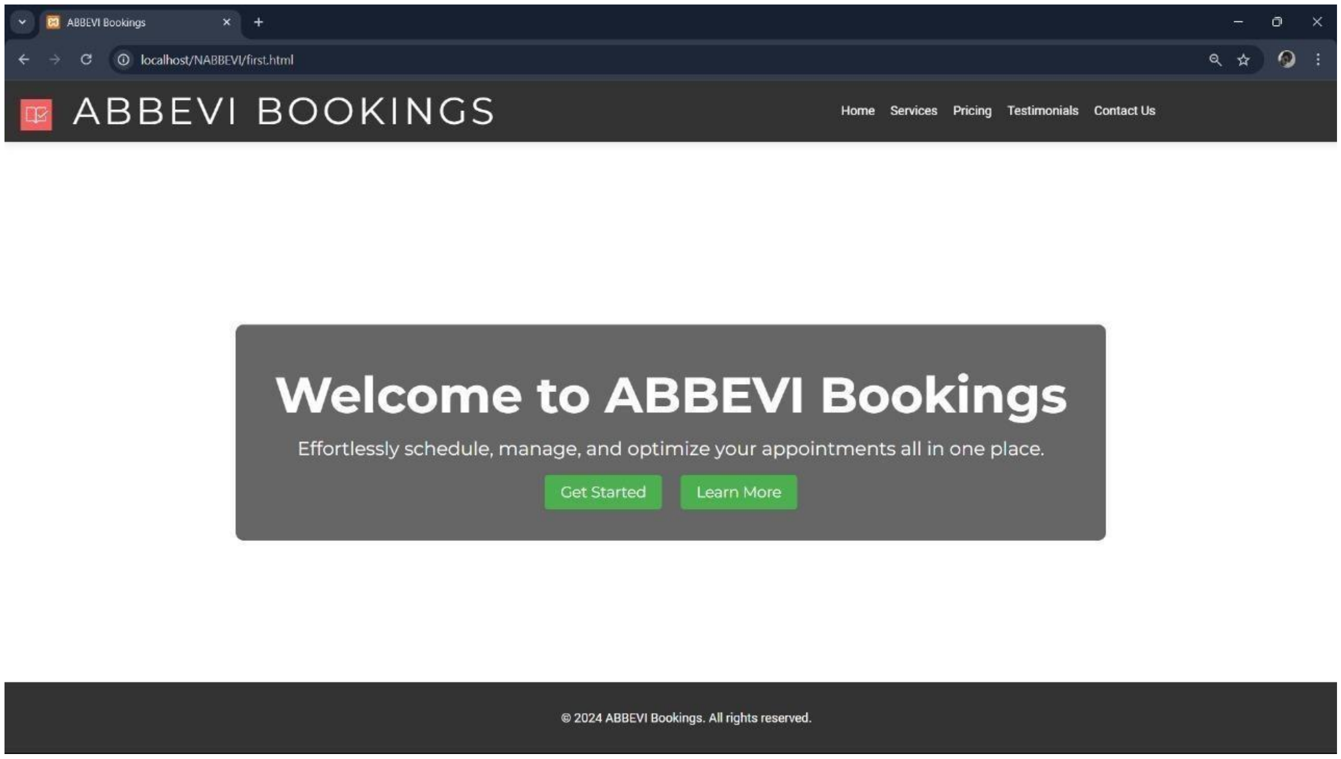
```
.footer-links a {
  color: #fff;
  margin: 0 10px;
  text-decoration: none;
}

.footer-links a:hover
  { color: #4CAF50;
}

/* Live Chat */
.live-chat {
  position: fixed;
  bottom:  20px;
  right: 20px;
  background-color:
  #4CAF50; color: #fff;
  padding: 10px 15px;
  border-radius:  30px;
  cursor: pointer;
  transition: transform 0.3s ease;
}

.live-chat:hover {
  transform: scale(1.1);
}
```

6.4 SCREENSHOTS

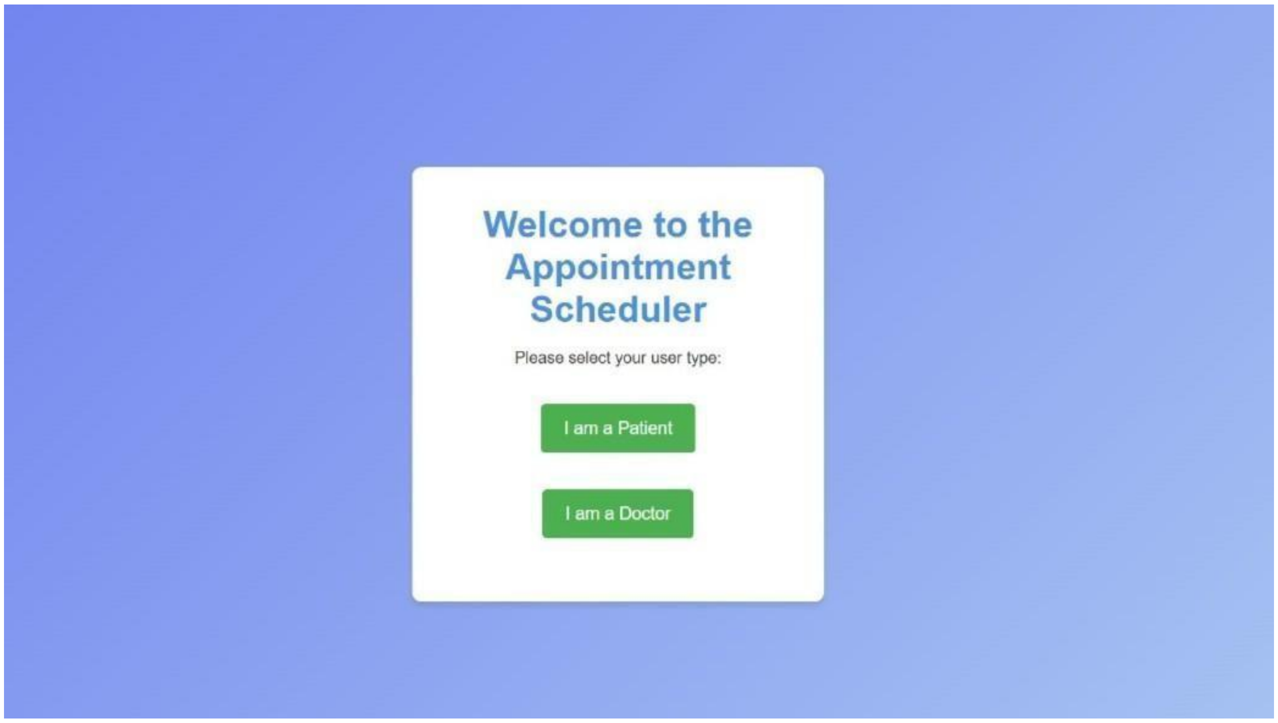
LOGIN PAGE


Login - Appointment Scheduler
localhost/NABBEVI/login_patient.php?
ABBEVI Bookings
Home
Sign in to ABBEVI Bookings
Email or phone
Enter your password
Remember Me
Forgot Password?
Login
Don't have an account? Sign up here
Privacy Policy
Terms of Service
Contact

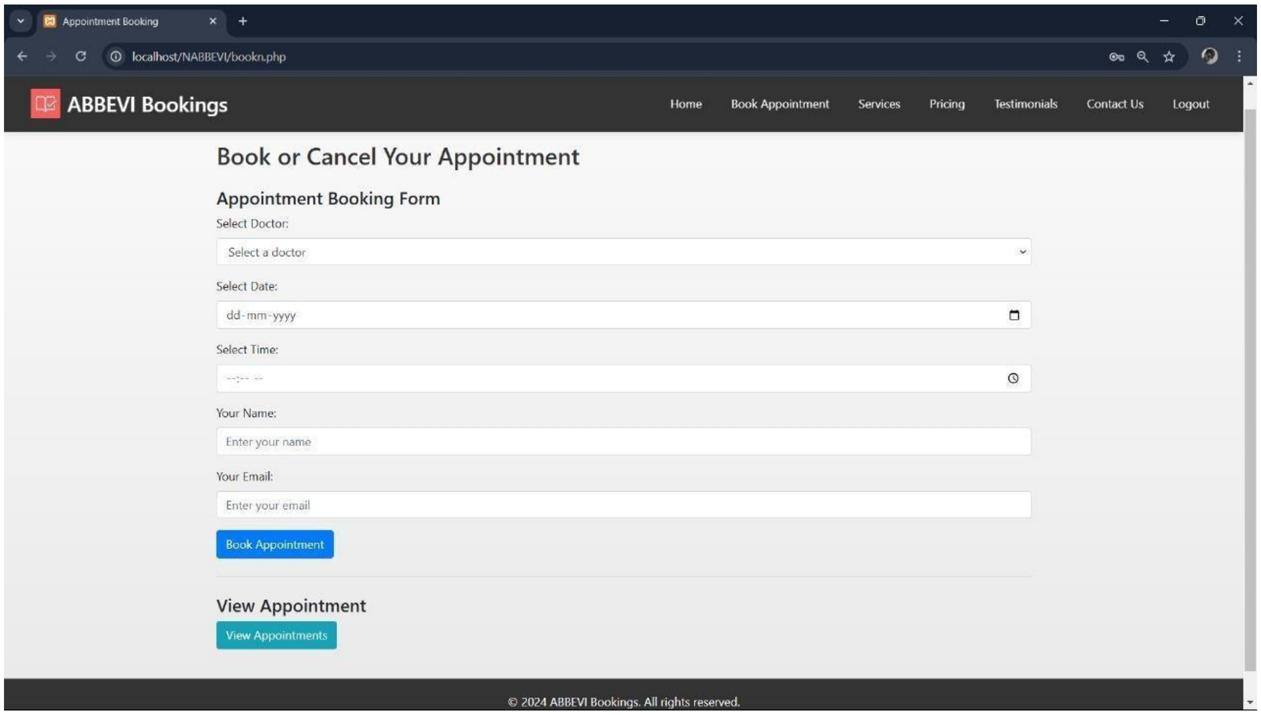
ADMIN HOME


ABBEVI Bookings
localhost/NABBEVI/first.html
ABBEVI BOOKINGS
Home
Services
Pricing
Testimonials
Contact Us
Welcome to ABBEVI Bookings
Effortlessly schedule, manage, and optimize your appointments all in one place.
Get Started
Learn More
© 2024 ABBEVI Bookings. All rights reserved.

USER HOME


Welcome to the Appointment Scheduler
Please select your user type:
I am a Patient
I am a Doctor

BOOKING PAGE



Appointment Booking

localhost/NABBEVI/bookn.php

ABBEVI Bookings

Home Book Appointment Services Pricing Testimonials Contact Us Logout

Book or Cancel Your Appointment

Appointment Booking Form

Select Doctor: Select a doctor

Select Date: dd-mm-yyyy

Select Time: --:-- --

Your Name: Enter your name

Your Email: Enter your email

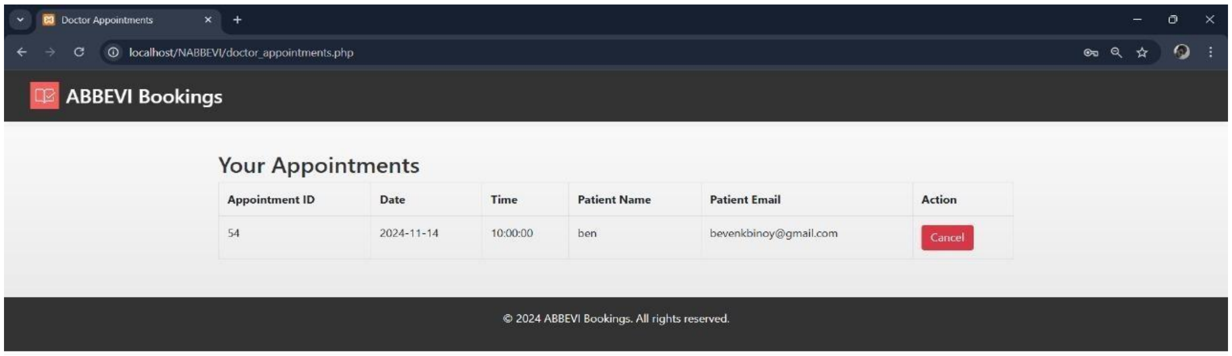
Book Appointment

View Appointment

View Appointments

© 2024 ABBEVI Bookings. All rights reserved.

APPOINMTENTS PAGE



Doctor Appointments

localhost/NABBEVI/doctor_appointments.php

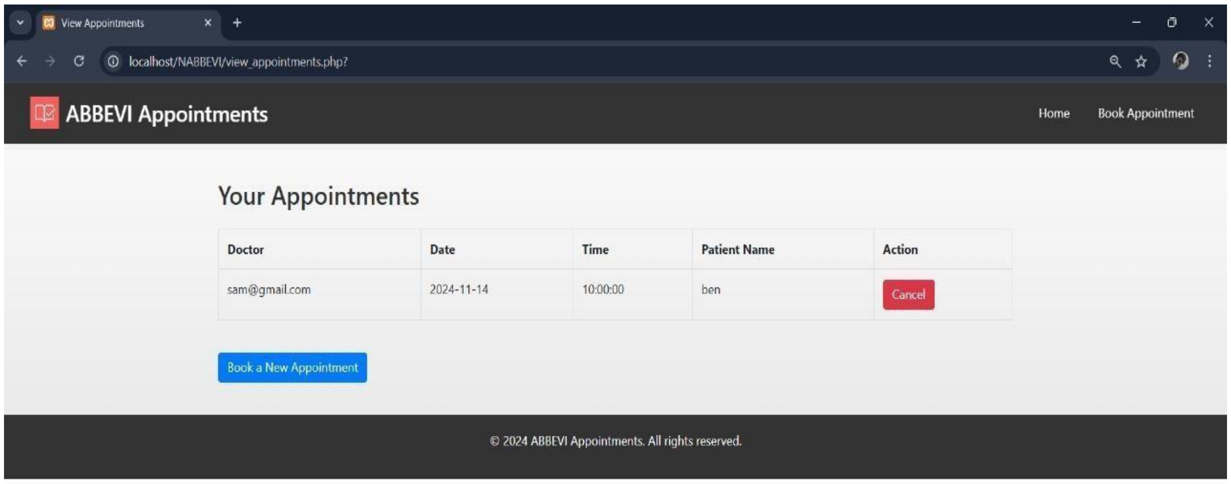
ABBEVI Bookings

Your Appointments

Appointment ID	Date	Time	Patient Name	Patient Email	Action
54	2024-11-14	10:00:00	ben	bevenkbinoy@gmail.com	Cancel

© 2024 ABBEVI Bookings. All rights reserved.

DOCTOR APPOINTMENT PAGE

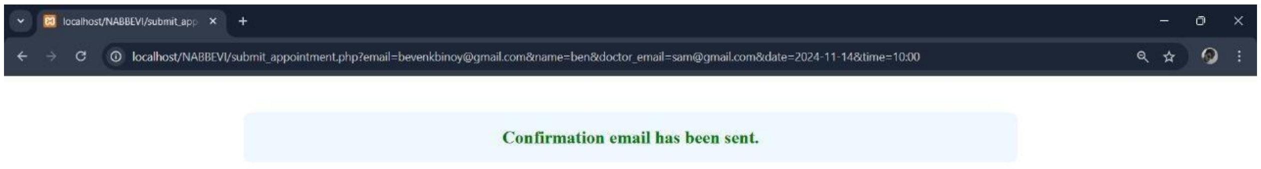


Your Appointments

Doctor	Date	Time	Patient Name	Action
sam@gmail.com	2024-11-14	10:00:00	ben	Cancel

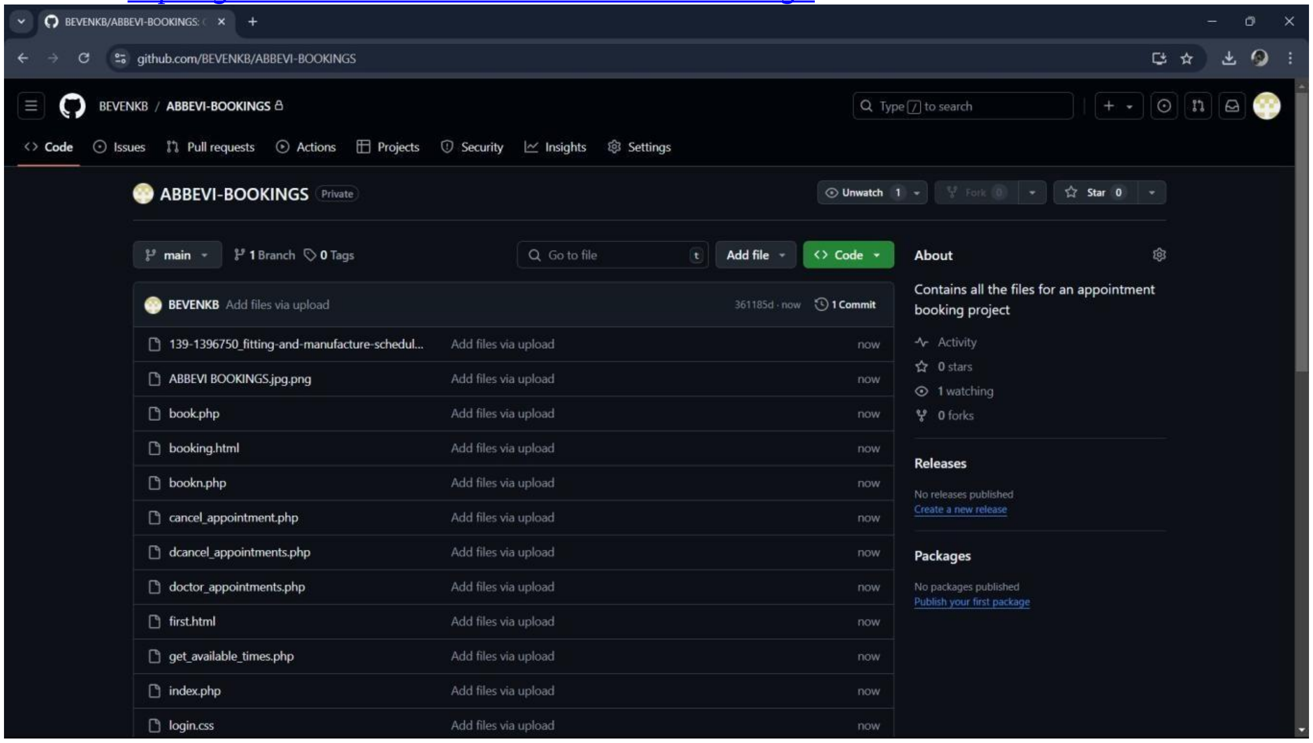
© 2024 ABBEVI Appointments. All rights reserved.

CONFIRMATION PAGE


localhost/NABBEVI/submit_app
localhost/NABBEVI/submit_appointment.php?email=bevenkbinoy@gmail.com&name=ben&doctor_email=sam@gmail.com&date=2024-11-14&time=10:00

Confirmation email has been sent.

GitHub: *https://github.com/BEVENKB/ABBEVI-BOOKINGS.git*



github.com/BEVENKB/ABBEVI-BOOKINGS

BEVENKB / ABBEVI-BOOKINGS

ABBEVI-BOOKINGS Private

main · 1 Branch · 0 Tags

BEVENKB Add files via upload · 361185d · now · 1 Commit

139-1396750_fitting-and-manufacture-schedul...	Add files via upload	now
ABBEVI BOOKINGS.jpg.png	Add files via upload	now
book.php	Add files via upload	now
booking.html	Add files via upload	now
bookn.php	Add files via upload	now
cancel_appointment.php	Add files via upload	now
dcancel_appointments.php	Add files via upload	now
doctor_appointments.php	Add files via upload	now
first.html	Add files via upload	now
get_available_times.php	Add files via upload	now
index.php	Add files via upload	now
login.css	Add files via upload	now

About

Contains all the files for an appointment booking project

Activity · 0 stars · 1 watching · 0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

CONCLUSION & FUTURE SCOPE

7.1 CONCLUSION

An effective appointment scheduler can significantly streamline operations, improve efficiency, and enhance customer satisfaction. By automating scheduling processes, businesses can reduce manual effort, minimize scheduling conflicts, and provide a more convenient experience for both service providers and clients. Additionally, features like automated reminders, calendar integration, and payment processing can further optimize workflows and increase revenue. As technology continues to evolve, advanced appointment scheduling solutions will offer even greater flexibility, scalability, and customization options, enabling businesses to stay competitive and deliver exceptional service.

7.2 FUTURE SCOPE

The future of appointment scheduling holds immense potential for innovation and improvement. As technology advances, we can anticipate several exciting developments. Artificial intelligence (AI) can revolutionize scheduling by predicting cancellations, optimizing time slots, and offering personalized recommendations. Integrating with other tools like CRM and marketing automation platforms can streamline operations and enhance customer engagement. Additionally, virtual and augmented reality can provide immersive appointment experiences, especially in healthcare and remote consultations. By embracing these emerging trends, appointment schedulers can become even more efficient, user-friendly, and valuable tools for businesses and individuals alike.

Additional Features (For Future Development):

- **Third-Party Calendar Integration**: Enable synchronization with services like Google Calendar, Outlook, etc.
- **Advanced Analytics**: Allow for more granular reporting (e.g., appointment statistics by day/week/month,

provider performance analysis).

- **Mobile App Integration**: Develop a mobile version of the platform for on-the-go scheduling and notifications.

REFERENCES

8.1 REFERENCES

- **PHP and MySQL Web Development**, Luke Welling & Laura Thomson - *5th Edition*, Addison-Wesley, 2016.
(This book covers PHP and MySQL development for building dynamic web applications, which is key for your backend logic and database management.)

- **JavaScript: The Good Parts**, Douglas Crockford, *O'Reilly Media*, 2008.
(This book provides an in-depth understanding of JavaScript, which is used to enhance interactivity in your scheduling system.)

- **PHP & MySQL: Novice to Ninja**, Tom Butler, *SitePoint*, 2017.
(This is a comprehensive guide for using PHP and MySQL together, making it ideal for understanding backend

development and database management.)

- **HTML and CSS: Design and Build Websites**, Jon Duckett, *Wiley*, 2011. *(A well-regarded resource for learning HTML and CSS to create clean, responsive, and intuitive front-end designs for your user interface.)*

- **JavaScript Enlightenment**, Cody Lindley, *First Edition*, based on JavaScript 1.5, ECMA-262, Edition. *(A detailed guide to JavaScript that provides a solid foundation for building dynamic features in web applications.)*

- **CSS Secrets: Better Solutions to Everyday Web Design Problems**, Lea Verou, *O'Reilly Media*, 2015. *(This book gives practical advice and techniques for mastering CSS to improve the visual layout and responsiveness of your application.)*

- **MySQL Reference Manual**, Oracle Corporation, *MySQL 8.0* - available online at: https://dev.mysql.com/doc/
(The official reference manual for MySQL, providing the most up-to-date information on database management and query syntax.)

- **Web Development with Node and Express**, Ethan Brown, *O'Reilly Media*, 2019. *(While this focuses on Node.js, it provides valuable insight into building scalable web applications, which can be useful for understanding backend frameworks.)*