Table of Contents

1. Introduction

1.1. Overview

In today’s fast-paced world, technology has become an indispensable part of our lives, especially in healthcare. With the rise of digital tools, healthcare services have seen significant advancements, enabling better patient care, personalized services, and efficient management. However, managing healthcare-related information remains a challenge for many individuals, especially for those with chronic conditions, requiring ongoing monitoring and regular medical check-ups.

This project, titled **AI Healthcare Assistant**, aims to bridge this gap by creating a web-based application that provides an AI-powered healthcare assistant for managing medical records, appointments, prescriptions, and health goals. This project aims to provide a personalized experience where patients can easily track their health, interact with an AI chatbot for assistance, and have all their medical data in one place. The system uses PHP for the backend and MySQL for the database, ensuring a smooth and responsive experience for the user.

The system allows users to upload and manage their medical reports and prescriptions, schedule and track appointments, and set health-related goals. The core feature of the system is an AI-powered healthcare assistant chatbot that interacts with users, answers health-related queries, and updates the user's reference file with every interaction. This reference file serves as a personalized repository of information, used to enhance future interactions with the chatbot, making the system smarter over time.

1.2. Objective of the Project

The main objective of this project is to develop a web application that integrates healthcare management features with an AI-powered chatbot to assist users in managing their health and medical records. The system aims to achieve the following objectives:

- Provide a user-friendly interface for managing healthcare records such as medical reports, prescriptions, and appointments.
- Offer personalized assistance to users through an AI chatbot, capable of answering queries and providing health-related advice.
- Implement a reference file system where each user’s medical history, health goals, and interactions are stored and used to improve future chatbot responses.
- Ensure secure user authentication and privacy by using session management with PHP and MySQL for user data storage.
- Integrate health goal tracking, appointment scheduling, and medical report management into a cohesive system.

1.3. Scope of the Project

The scope of this project includes the development of a complete web-based system that allows users to interact with an AI-powered chatbot for health-related queries. The system also enables users to upload medical reports, prescriptions, track their health goals, and manage doctor appointments.

Key features of the system include:

- **User Authentication**: Secure login and registration system using PHP sessions and MySQL for storing user credentials and session data.
- **AI Healthcare Assistant**: A chatbot that helps users with health-related queries and stores interaction data for future reference.
- **Medical Report and Prescription Management**: Allows users to upload and manage their medical reports and prescriptions.
- **Health Goal Management**: Users can set, track, and manage their health goals, with progress updates and reminders.
- **Appointment Management**: Enables users to schedule, view, and manage appointments with healthcare professionals.

The project is built using PHP, MySQL, HTML, CSS, and JavaScript. The backend is powered by PHP, which communicates with the MySQL database to manage user data, medical records, health goals, and appointments. The frontend utilizes HTML, CSS, and JavaScript to provide a responsive and interactive user interface. The chatbot functionality is integrated with AI-based algorithms that simulate a human-like conversation and provide intelligent responses to user queries.

1.4. Tools and Technologies Used

The development of this project utilizes several modern technologies, including:

- **PHP**: A server-side scripting language used to handle the backend logic, such as user authentication, database interactions, and the chatbot functionality.
- **MySQL**: A relational database management system used for storing user data, medical records, prescriptions, appointments, and health goals.
- **HTML/CSS**: These technologies are used for creating the structure and styling of the web application, ensuring it is responsive and user-friendly.
- **JavaScript**: JavaScript is used for front-end interactivity, including form validation and dynamic content updates on the user interface.
- **AI Integration (Gemini API)**: The AI-powered chatbot is integrated with the Gemini API, allowing it to simulate a real-time conversation, answering health-related queries and updating user reference files after each interaction.

By using these technologies, the project provides an intuitive, secure, and functional solution for managing healthcare information, empowering users to take charge of their health in a more efficient and interactive way.

2. Profile of the Problem

2.1. Problem Statement

Healthcare has always been one of the most crucial aspects of human well-being. However, despite advancements in technology, managing personal health has become increasingly complex due to the rise of chronic diseases, multiple health check-ups, medical records, and the need for continuous monitoring. Many individuals struggle with managing their healthcare efficiently, often losing track of vital health information, failing to monitor their progress on health goals, or forgetting important medical appointments. Moreover, the growing demand for personalized healthcare services calls for innovative solutions that allow individuals to access their health records, prescriptions, and medical history anytime and anywhere.

One of the primary challenges in the healthcare industry is the lack of effective and centralized management of medical records. Patients are often required to store and manage numerous reports, prescriptions, and appointments in physical or digital formats, which can be cumbersome and time-consuming. Without a single platform to manage all aspects of their health, individuals face difficulties in keeping track of their health status and medical needs.

Another challenge is the lack of personalized care. While many healthcare facilities provide medical services, there is often a gap in delivering tailored advice to patients based on their medical history and ongoing treatment plans. A lack of personalized interaction means patients may miss out on relevant health information or advice that could improve their outcomes. Additionally, in traditional healthcare settings, patients may have to wait for hours for appointments or rely on fragmented health systems to communicate their symptoms or concerns.

This project addresses these challenges by providing a centralized platform that integrates an AI healthcare assistant with functionalities such as medical record management, appointment scheduling, health goal tracking, and AI-powered personalized interactions. The application provides a user-friendly interface to store, manage, and update medical data, making it easy for users to keep track of their health in one place. The AI chatbot, which acts as a healthcare assistant, offers personalized health advice and updates the reference file with every new interaction, helping patients make informed decisions about their health.

2.2. Existing System

Currently, several healthcare systems are available that provide some level of medical record management and appointment scheduling. However, most of these systems are not fully integrated or lack user-friendly features, making them difficult for average users to interact with efficiently. Some of the existing healthcare management systems are:

- **MyChart**: A widely used patient portal that allows users to manage appointments, view test results, and communicate with healthcare providers. While it offers a good system for managing records and scheduling appointments, it lacks an intelligent system to assist users with medical queries and does not offer personalized health advice.
- **Medisafe**: An app that helps users manage their medications and prescriptions, offering reminders and detailed tracking. However, it does not integrate with health goals or provide a broader system for managing general health, medical reports, and doctor appointments.
- **HealthVault**: A platform that enables users to store their health information digitally. However, it does not offer a personalized AI assistant or the capability to update the system based on real-time data or user interactions.

These systems, while functional, typically focus on a narrow range of services, often lacking features like real-time interaction, AI assistance, and personalized health management. In addition, many users may find it cumbersome to interact with multiple apps or platforms to manage different aspects of their health. Furthermore, they often fail to centralize health records, resulting in a fragmented experience where users have to rely on multiple sources for different types of health data.

The existing systems also lack an effective method of managing and tracking health goals over time. Patients with chronic conditions or long-term health objectives need a tool that helps them set, track, and evaluate their progress toward these goals. Without such tools, patients are likely to lose motivation or forget about their health targets.

2.3. Gap Analysis

The current landscape of healthcare management tools reveals a significant gap in terms of user-centered, AI-driven solutions. While existing systems provide basic functionality for record-keeping and scheduling, they fail to offer personalized healthcare assistance or integrate health data from multiple sources into a unified system. Additionally, there is a lack of systems that empower users to track and manage their health goals, which is a critical aspect of long-term well-being.

Furthermore, the systems available today are often not easily accessible or usable by all patients, especially those who may not be tech-savvy or are unfamiliar with complex medical systems. This limits the widespread adoption of these tools and their potential to improve patient outcomes.

This project aims to address these gaps by creating a comprehensive healthcare management platform that integrates AI, medical report storage, health goal tracking, and appointment scheduling into one cohesive system. By combining these features with personalized interaction via an AI chatbot, the system will offer a more user-friendly and effective solution for managing personal health. The chatbot will be able to guide users through their healthcare journey, answer their queries, and ensure that their health data is consistently updated and accessible.

2.4. Rationale/Scope of Study

The rationale behind this project is to improve how patients manage and track their healthcare. The integration of AI into healthcare systems offers immense potential to enhance the overall user experience and provide tailored recommendations based on individual health histories and ongoing treatment plans. By leveraging technologies such as PHP, MySQL, and AI, this project will provide users with an intuitive platform for managing their health.

This platform will cater to individuals seeking a comprehensive, user-friendly, and personalized healthcare experience. The key benefits of this system include:

- **Centralized Health Records**: Users can store and manage all their health-related information in one place, including reports, prescriptions, appointments, and health goals.
- **Personalized AI Assistance**: The chatbot will offer personalized guidance, making healthcare information more accessible and relevant to the user.
- **Health Goal Tracking**: The system will enable users to set and track health goals, helping them stay motivated and aligned with their wellness objectives.
- **Appointment Management**: Users can easily schedule, view, and manage appointments with healthcare providers.

This system aims to make healthcare management easier, more efficient, and accessible to individuals of all ages and tech proficiency, ultimately contributing to better health outcomes.

Here is a 2-page "Problem Analysis" section for your project report:

---

3. Problem Analysis

3.1. Product Definition

The product in question is a comprehensive AI-powered healthcare assistant system designed to help users manage their personal health data, track health goals, and schedule medical appointments efficiently. The system provides a centralized platform for users to upload and store medical reports, prescriptions, and other important health records. Moreover, it includes an AI chatbot that serves as a healthcare assistant, offering personalized health advice, answering user queries, and maintaining a dynamic reference file for each user. This reference file is updated in real-time as users interact with the system, ensuring that the chatbot provides the most relevant advice based on the user's ongoing health status, goals, and appointments.

The key features of the system include:

- **User Authentication and Session Management**: Secure login functionality using PHP and MySQL to ensure that only authenticated users can access their health information.
- **AI Healthcare Chatbot**: A chatbot powered by AI that provides personalized health advice, updates user reference files, and interacts with users based on their health data and goals.
- **Health Record Management**: Users can upload and store medical reports, prescriptions, and other health-related documents in a centralized repository.
- **Health Goals and Tracking**: The system allows users to create, manage, and track their health goals, helping them stay on top of their well-being.
- **Appointment Management**: The application offers an appointment scheduling system, enabling users to view, manage, and schedule doctor appointments efficiently.

The system is designed to provide a simple yet effective solution for individuals who are seeking to take control of their health by managing their medical records, setting health goals, and receiving personalized health guidance. It eliminates the need for multiple applications and physical paperwork, offering a unified platform for all aspects of healthcare management.

3.2. Feasibility Analysis

The feasibility of this project can be evaluated from three primary perspectives: technical feasibility, operational feasibility, and economic feasibility.

3.2.1. Technical Feasibility

From a technical standpoint, the project is highly feasible. The core technologies used in this project are PHP for backend development, MySQL for database management, and AI for chatbot functionality. All of these technologies are well-established, with extensive documentation and support available. PHP and MySQL are widely used for building web applications, making them reliable choices for creating the backend infrastructure. AI integration through platforms such as Gemini or other language models is feasible with the right API integration, which will allow the chatbot to interact with users in a conversational manner, providing them with relevant health-related information.

The platform’s design uses common web technologies such as HTML, CSS, JavaScript, and PHP, making it compatible with a wide range of devices and browsers. PHP and MySQL are capable of handling the user authentication, session management, and database operations required to store and retrieve user health data efficiently. Additionally, the AI-powered chatbot is implemented in a way that updates the reference files dynamically based on user interactions, ensuring that users get personalized and accurate information.

3.2.2. Operational Feasibility

The operational feasibility of this project is also high. The project will be deployed on a web server using PHP and MySQL, ensuring that it can be accessed from any device with an internet connection. Users will have the ability to interact with the chatbot, upload reports and prescriptions, and track their health goals from a centralized dashboard. Since the system is built with PHP and MySQL, the backend is easily manageable and maintainable, with straightforward data operations such as storing, retrieving, and updating user information.

The AI healthcare assistant is designed to function within the capabilities of modern web servers, ensuring that the application runs smoothly and provides timely responses to user inputs. Additionally, the platform's functionality is not dependent on any high-end hardware, making it accessible to a wide user base. The system can also be scaled as needed to accommodate a growing number of users or additional features.

3.2.3. Economic Feasibility

From an economic perspective, this project is cost-effective to develop and deploy. The technologies used, such as PHP, MySQL, and AI services, are cost-efficient, and the tools for development and deployment (e.g., XAMPP, Netlify) are available for free or at a low cost. The server infrastructure required for hosting the system is affordable, and open-source AI models or third-party API services can be used for the chatbot functionality without significant investment.

Moreover, the system is designed to offer a range of features that add value to the users. The platform enables individuals to manage their health efficiently, track health goals, and receive personalized health advice, all of which are essential in promoting better healthcare practices. By consolidating all these functions into one platform, users can avoid the need for multiple services and applications, resulting in cost savings for both the users and the healthcare system.

3.3. Project Plan

The development of this project can be divided into several key phases, each contributing to the overall success of the system:

3.3.1. Phase 1: Requirements Gathering and Analysis

- Conduct a detailed analysis of the requirements for the AI healthcare assistant system.
- Identify user needs, health-related features, and chatbot functionality.
- Design the user interface and database schema.

3.3.2. Phase 2: Design and Development

- Develop the front-end user interface using HTML, CSS, and JavaScript.
- Set up the PHP backend and integrate MySQL for user authentication, health record management, and data storage.
- Implement the AI chatbot functionality using third-party APIs or custom models.

3.3.3. Phase 3: Testing

- Perform extensive testing of the system to ensure all features work as expected.
- Test the AI chatbot to ensure it provides accurate and personalized responses.
- Perform user acceptance testing to ensure the platform is intuitive and user-friendly.

3.3.4. Phase 4: Deployment

- Deploy the system on a live server.
- Integrate additional features such as appointment scheduling and health goal tracking.

- Conduct final testing in a production environment.

3.3.5. Phase 5: Maintenance and Updates

- Monitor the system for issues and bugs.
- Release regular updates to improve performance and introduce new features.

3.4. Flowcharts and Pseudo Code

Flowcharts and pseudo code will be developed to represent the logic and workflow of key features in the system. The flowcharts will visually demonstrate the process of user authentication, chatbot interactions, and health record management, while the pseudo code will outline the logical steps behind these processes.

5. Implementation

The implementation phase of this project involved the development and integration of various components to create the AI-powered healthcare assistant system. This section outlines the steps taken during the implementation process, the challenges encountered, and the solutions applied.

5.1. Implementation of the Project

The implementation of the AI-powered healthcare assistant involved building the backend infrastructure using PHP, MySQL, and integrating an AI-powered chatbot for interactive communication with users. The system was designed to manage user authentication, handle health-related data, and enable a chatbot to interact with the users based on their health information. The following sections describe the key features and their implementation in detail.

5.1.1. User Authentication and Session Management

To ensure secure access to the system, user authentication was implemented using PHP and MySQL. A MySQL database was used to store user information, such as login credentials (username and password). The user authentication process involves validating the credentials entered during login against the data stored in the MySQL database. If the credentials match, the user is granted access to the system.

The user session is managed using PHP sessions, ensuring that the user remains logged in throughout their interaction with the system. The session also helps in identifying the user and linking their data (such as health records and goals) to the correct profile. Session variables were used to store the user's login status and ID, which are checked before allowing access to various pages.

5.1.2. AI Healthcare Chatbot Integration

The core feature of this project is the AI healthcare assistant, which uses a chatbot to provide personalized health advice to users. The chatbot was integrated using third-party AI models or APIs. The chatbot interacts with users based on their health data stored in the system, such as medical reports, prescriptions, health goals, and appointments.

The chatbot is designed to analyze the user's queries and provide responses that are contextually relevant. It accesses the user's reference file, which is a dynamic document that is updated every time the user interacts with the system (e.g., adding a report, updating a health goal, or completing a chat). This reference file serves as a knowledge base, allowing the chatbot to offer advice based on the user's medical history and health goals.

The chatbot was implemented using a combination of PHP for backend processing and a third-party AI service for natural language processing (NLP). The integration was achieved through API calls, where the system sends user queries to the AI service and retrieves responses, which are then displayed to the user in the chat interface.

5.1.3. Health Record Management

The health record management system allows users to upload and store medical reports, prescriptions, and other health-related documents. This feature is implemented using PHP for backend handling and MySQL for data storage. When a user uploads a file, it is stored in a specific folder on the server, and the details (e.g., file name, upload date, and user ID) are saved in the MySQL database. This information is then accessible from the user's dashboard.

Users can view, update, and delete their reports and prescriptions, and the system ensures that only the authenticated user can access their health records. The health records are linked to the user's reference file, and the chatbot uses this data to provide relevant responses. For example, if the user asks about a specific condition, the chatbot can refer to the user's medical reports to provide an informed response.

5.1.4. Health Goals and Tracking

The health goals feature allows users to create, manage, and track their health goals. Each user can define their goals, such as losing weight, increasing physical activity, or controlling blood sugar levels. These goals are stored in the MySQL database and are linked to the user's profile.

Users can update the status of their goals, and the system tracks their progress over time. The AI healthcare assistant uses this information to provide motivational advice, progress updates, and suggestions on how to achieve the user's health goals. The health goals are also integrated into the user's reference file, ensuring that the chatbot provides advice based on the current status of the goals.

5.1.5. Appointment Management System

The appointment management feature enables users to schedule, view, and manage doctor appointments. This feature is implemented using PHP for backend handling and MySQL for data storage. The user can input appointment details, including the doctor's name, date, time, and reason for the visit. This information is stored in the MySQL database and is displayed on the user's dashboard.

The system allows users to view upcoming appointments, cancel or reschedule them, and track their appointment history. The AI chatbot can also refer to the user's appointment history when providing health advice. For example, if a user asks about a follow-up visit to a doctor, the chatbot can provide details about the next appointment based on the user's records.

5.2. Conversion Plan

Once the core functionality of the system was developed, the next step was to ensure that the system was compatible with different devices and browsers. The application was designed to be responsive, ensuring that users can access it from both desktop and mobile devices. The frontend was developed using HTML, CSS, and JavaScript to create a user-friendly interface that is intuitive and easy to navigate.

The backend, implemented with PHP and MySQL, was designed to be scalable, so that additional features could be added as the project evolves. The chatbot integration was tested extensively to ensure accurate and relevant responses based on the user's health data.

5.3. Post-Implementation and Software Maintenance

After the deployment of the system, ongoing maintenance is necessary to ensure the continued smooth functioning of the application. This includes fixing any bugs that arise, updating the AI model to improve its responses, and adding new features based on user feedback. Regular backups of the MySQL database are taken to prevent data loss, and security measures such as encryption and secure authentication are applied to protect user data.

The system is also continuously monitored for performance issues, and optimization techniques are implemented to ensure that the platform remains fast and responsive even as the user base grows. Additionally, the platform will be updated with new health-related features, integrating with external health data sources and APIs to provide users with even more personalized healthcare assistance.

Source Code:

Db.php

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

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}
?>
```

Create_tables.php

```php
<?php
```

```
// Include database connection file
include 'db.php';

// SQL to create 'users' table
$sql_users = "CREATE TABLE IF NOT EXISTS users (
    id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    firstname VARCHAR(50) NOT NULL,
    lastname VARCHAR(50) NOT NULL,
    email VARCHAR(50) NOT NULL UNIQUE,
    phone VARCHAR(15) NOT NULL,
    dob DATE NOT NULL,
    password VARCHAR(255) NOT NULL,
    reg_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
)";

if ($conn->query($sql_users) !== TRUE) {
    echo "Error creating 'users' table: " . $conn->error . "<br>";
} else {
    echo "'users' table created or already exists.<br>";
}

// SQL to create 'health_goals' table
$sql_goals = "CREATE TABLE IF NOT EXISTS health_goals (
    id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    user_id INT(6) UNSIGNED NOT NULL,
    goal_type VARCHAR(50) NOT NULL,
    goal_title VARCHAR(100) NOT NULL,
    target TEXT NOT NULL,
    timeline DATE NOT NULL,
    reminders TEXT,
    notes TEXT,
    progress INT(3) DEFAULT 0,
    status VARCHAR(50) DEFAULT 'Not Started',
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE
)";

if ($conn->query($sql_goals) !== TRUE) {
    echo "Error creating 'health_goals' table: " . $conn->error . "<br>";
} else {
    echo "'health_goals' table created or already exists.<br>";
}

// SQL to create 'reports' table
$sql_reports = "CREATE TABLE IF NOT EXISTS reports (
    id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    user_id INT(6) UNSIGNED NOT NULL,
    doctor VARCHAR(255) NOT NULL,
    file_name VARCHAR(255) NOT NULL,
    file_path VARCHAR(255) NOT NULL,
    file_type VARCHAR(50) NOT NULL,
    upload_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE
)";

if ($conn->query($sql_reports) !== TRUE) {
    echo "Error creating 'reports' table: " . $conn->error . "<br>";
} else {
    echo "'reports' table created or already exists.<br>";
}

// SQL to create 'appointments' table
$sql_appointments = "CREATE TABLE IF NOT EXISTS appointments (
    id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    user_id INT(6) UNSIGNED NOT NULL,
    doctor_name VARCHAR(100) NOT NULL,
    appointment_date DATE NOT NULL,
    appointment_time TIME NOT NULL,
    notes TEXT,
    status VARCHAR(50) DEFAULT 'Scheduled',
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE
)";

if ($conn->query($sql_appointments) !== TRUE) {
    echo "Error creating 'appointments' table: " . $conn->error . "<br>";
} else {
    echo "'appointments' table created or already exists.<br>";
}

// SQL to create 'prescriptions' table
$sql_prescriptions = "CREATE TABLE IF NOT EXISTS prescriptions (
    id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    user_id INT(6) UNSIGNED NOT NULL,
    file_name VARCHAR(255) NOT NULL,
    file_path VARCHAR(255) NOT NULL,
    file_type VARCHAR(50) NOT NULL,
    upload_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE
)";

if ($conn->query($sql_prescriptions) !== TRUE) {
    echo "Error creating 'prescriptions' table: " . $conn->error . "<br>";
} else {
    echo "'prescriptions' table created or already exists.<br>";
}

?>
```

Dashboard.php

```
<?php
session_start(); // Start the session

// Check if the user is logged in
if (!isset($_SESSION['user_id'])) {
  // Redirect to login page if not logged in
  header("Location: login_signup.php");
  exit();
}

// Access user data from session
$user_id = $_SESSION['user_id'];
$user_email = $_SESSION['user_email'];
$user_phone = $_SESSION['user_phone'];
?>
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>AI Dashboard</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet" />
  <link href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.1/font/bootstrap-icons.css" rel="stylesheet" />
  <link href="assets/css/styles.css" rel="stylesheet" />
</head>

<body>
  <div class="wrapper">
    <nav id="sidebar" class="active">
      <div class="sidebar-header">
        <h3>AI Dashboard</h3>
      </div>
      <ul class="list-unstyled components">
        <li class="active">
          <a href="dashboard.php"><i class="bi bi-chat-dots-fill me-2"></i>Chat</a>
        </li>
        <li>
          <a href="dashboard.php#features"><i class="bi bi-grid-fill me-2"></i>Features</a>
        </li>
        <li>
          <a href="profile.php"><i class="bi bi-person-fill me-2"></i>Profile</a>
        </li>
        <li>
          <a href="settings.php"><i class="bi bi-gear-fill me-2"></i>Settings</a>
        </li>
  <li>
   <a href="appointments.php"><i class="bi bi-calendar-check me-2"></i>Appointments</a>
  </li>
  <li>
   <a href="reports.php"><i class="bi bi-file-medical me-2"></i>Reports</a>
  </li>
  <li>
   <a href="goals.php"><i class="bi bi-trophy me-2"></i>Health Goals</a>
  </li>
 </ul>
</nav>

<div id="content">
 <nav class="navbar navbar-expand-lg navbar-light bg-light">
  <div class="container-fluid">
   <button type="button" id="sidebarCollapse" class="btn btn-primary">
    <i class="bi bi-list"></i>
   </button>
   <div class="ms-2">
    <input type="text" class="form-control" placeholder="Search..." />
   </div>
   <div class="ms-auto d-flex align-items-center">
    <div class="dropdown">
     <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown">
      <i class="bi bi-person-circle fs-5"></i>
     </a>
     <ul class="dropdown-menu dropdown-menu-end">
      <li>
       <a class="dropdown-item" href="profile.php">Profile</a>
      </li>
      <li>
       <a class="dropdown-item" href="settings.php">Settings</a>
      </li>
      <li>
       <hr class="dropdown-divider" />
      </li>
      <li><a class="dropdown-item" href="logout.php">Logout</a></li>
     </ul>
    </div>
   </div>
  </div>
 </nav>

 <div class="chat-container">
  <div class="chat-messages" id="chatMessages">
   <div class="message ai-message">
    <div class="message-content">
     <i class="bi bi-robot"></i>
     <div class="message-text">
      Hello! How can I assist you today?
```

```html
        </div>
      </div>
    </div>
    <div class="message user-message">
      <div class="message-content">
        <i class="bi bi-person"></i>
        <div class="message-text">
          Hi! Can you help me with some information?
        </div>
      </div>
    </div>
  </div>
  <div class="chat-input-container">
    <div class="input-group">
      <textarea class="form-control" placeholder="Type your message here..." rows="1"></textarea>
      <button class="btn btn-primary">
        <i class="bi bi-send"></i>
      </button>
    </div>
  </div>
</div>

<section id="features" class="features-section">
  <div class="container">
    <div class="section-title text-center mb-5">
      <h2>Our Features</h2>
      <p>Experience the power of AI-driven healthcare assistance</p>
    </div>

    <div class="row justify-content-center">
      <div class="col-md-8">
        <ul class="nav nav-tabs nav-fill mb-4" role="tablist">
          <li class="nav-item">
            <a class="nav-link active" data-bs-toggle="tab" href="#chat-assistant">
              <i class="bi bi-chat-dots me-2"></i>Smart Chat Assistant
            </a>
          </li>
          <li class="nav-item">
            <a class="nav-link" data-bs-toggle="tab" href="#intelligent-responses">
              <i class="bi bi-brain me-2"></i>Intelligent Responses
            </a>
          </li>
          <li class="nav-item">
            <a class="nav-link" data-bs-toggle="tab" href="#data-analysis">
              <i class="bi bi-graph-up me-2"></i>Data Analysis
            </a>
          </li>
        </ul>

        <div class="tab-content">
```

```html
<div class="tab-pane fade show active" id="chat-assistant">
  <div class="feature-content">
    <h3>Smart Chat Assistant</h3>
    <p class="text-muted">
      Experience natural conversations with our AI-powered
      chat assistant
    </p>
    <ul class="feature-list">
      <li>
        <i class="bi bi-check2-circle"></i> Natural language
        processing for human-like interactions
      </li>
      <li>
        <i class="bi bi-check2-circle"></i> 24/7 availability
        for instant responses
      </li>
      <li>
        <i class="bi bi-check2-circle"></i> Context-aware
        conversations for better understanding
      </li>
    </ul>
  </div>
</div>
<div class="tab-pane fade" id="intelligent-responses">
  <div class="feature-content">
    <h3>Intelligent Responses</h3>
    <p class="text-muted">
      Get accurate and contextual responses powered by
      advanced AI
    </p>
    <ul class="feature-list">
      <li>
        <i class="bi bi-check2-circle"></i> Machine
        learning-based response generation
      </li>
      <li>
        <i class="bi bi-check2-circle"></i> Adaptive learning
        from user interactions
      </li>
      <li>
        <i class="bi bi-check2-circle"></i> Multi-language
        support for global accessibility
      </li>
    </ul>
  </div>
</div>
<div class="tab-pane fade" id="data-analysis">
  <div class="feature-content">
    <h3>Data Analysis</h3>
    <p class="text-muted">
```

```
                      Leverage powerful analytics for deeper insights
                    </p>
                    <ul class="feature-list">
                      <li>
                        <i class="bi bi-check2-circle"></i> Real-time data
                        processing and analysis
                      </li>
                      <li>
                        <i class="bi bi-check2-circle"></i> Customizable
                        analytics dashboard
                      </li>
                      <li>
                        <i class="bi bi-check2-circle"></i> Trend
                        identification and reporting
                      </li>
                    </ul>
                  </div>
                </div>
              </div>
            </div>
          </div>
        </div>
      </section>
    </div>
  </div>

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>
  <script src="assets/js/script.js"></script>
</body>

</html>
```

Appointments.php

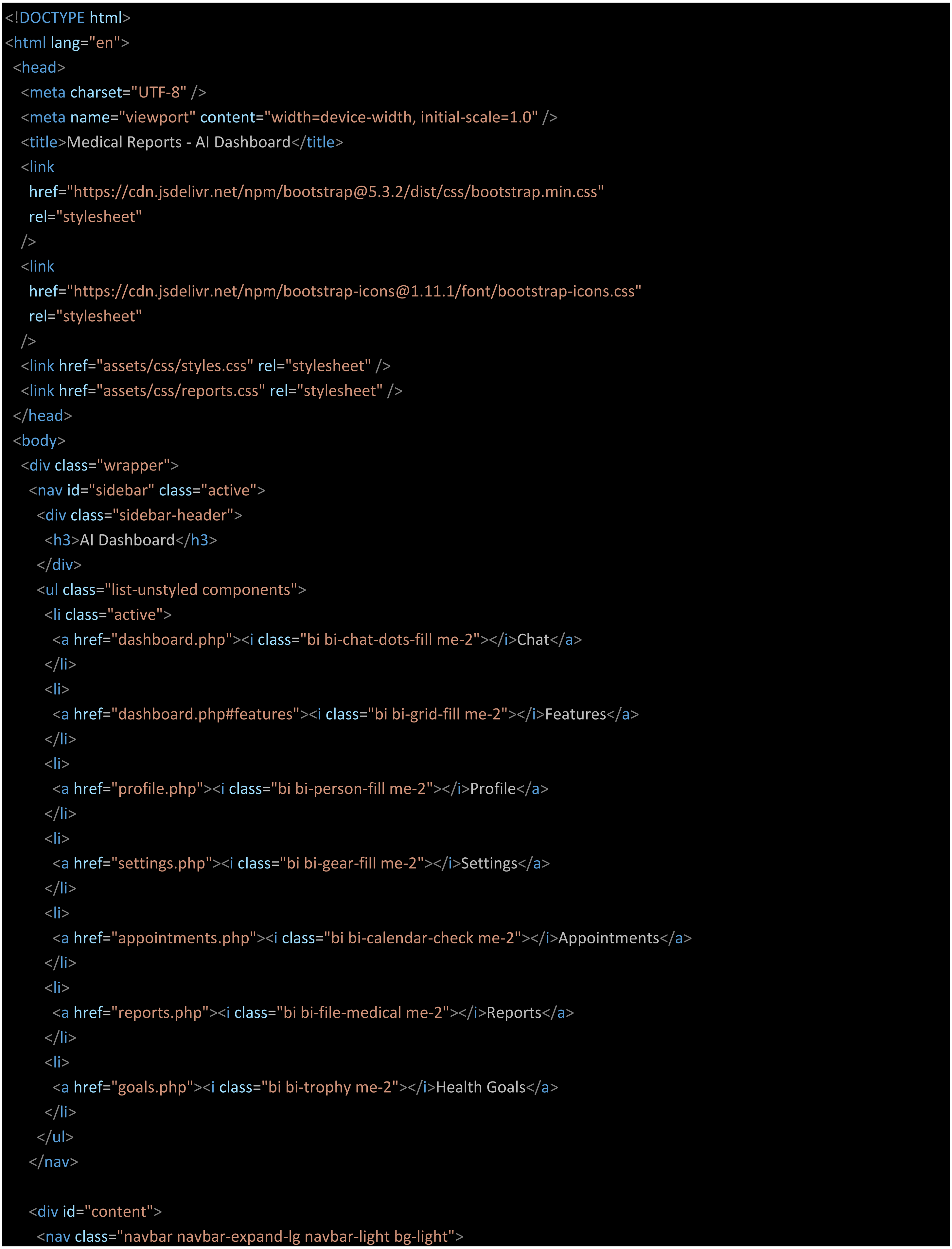
```
<?php
session_start();
include 'db.php'; // Database connection
include 'create_tables.php'; // Table creation script

// If the user is not logged in, redirect them to the login page
if (!isset($_SESSION['user_id'])) {
  echo json_encode(['success' => false, 'error' => 'User not logged in']);
  exit();
}

// Handle form submission to insert an appointment
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
  $user_id = $_SESSION['user_id']; // Get user ID from session
  $doctor_name = $_POST['doctor_name']; // Get doctor name from form
  $appointment_date = $_POST['appointment_date']; // Get appointment date from form
  $appointment_time = $_POST['appointment_time']; // Get appointment time from form
  $notes = $_POST['notes']; // Get notes from form

  // Prepare the SQL query to insert the appointment
  $stmt = $conn->prepare("INSERT INTO appointments (user_id, doctor_name, appointment_date, appointment_time, notes)
                          VALUES (?, ?, ?, ?, ?)");
  $stmt->bind_param("issss", $user_id, $doctor_name, $appointment_date, $appointment_time, $notes);

  // // Execute the query and check for errors
  // if ($stmt->execute()) {
  //   echo json_encode(['success' => true]);
  // } else {
  //   echo json_encode(['success' => false, 'error' => $stmt->error]);
  // }

  // Close the statement
  $stmt->close();
}
?>


<!-- Rest of your HTML code for displaying the sidebar and other UI elements -->


<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Appointments - AI Dashboard</title>
  <link
    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
    rel="stylesheet" />
  <link
    href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.1/font/bootstrap-icons.css"
    rel="stylesheet" />
  <link href="assets/css/styles.css" rel="stylesheet" />
  <link href="assets/css/appointments.css" rel="stylesheet" />
</head>

<body>
  <div class="wrapper">
    <nav id="sidebar" class="active">
      <div class="sidebar-header">
        <h3>AI Dashboard</h3>
      </div>
      <ul class="list-unstyled components">
        <li class="active">
      <a href="dashboard.php"><i class="bi bi-chat-dots-fill me-2"></i>Chat</a>
    </li>
    <li>
      <a href="dashboard.php#features"><i class="bi bi-grid-fill me-2"></i>Features</a>
    </li>
    <li>
      <a href="profile.php"><i class="bi bi-person-fill me-2"></i>Profile</a>
    </li>
    <li>
      <a href="settings.html"><i class="bi bi-gear-fill me-2"></i>Settings</a>
    </li>
    <li>
      <a href="appointments.php"><i class="bi bi-calendar-check me-2"></i>Appointments</a>
    </li>
    <li>
      <a href="reports.php"><i class="bi bi-file-medical me-2"></i>Reports</a>
    </li>
    <li>
      <a href="goals.php"><i class="bi bi-trophy me-2"></i>Health Goals</a>
    </li>
  </ul>
</nav>

<div id="content">
  <nav class="navbar navbar-expand-lg navbar-light bg-light">
    <div class="container-fluid">
      <button type="button" id="sidebarCollapse" class="btn btn-primary">
        <i class="bi bi-list"></i>
      </button>
      <div class="ms-2">
        <input type="text" class="form-control" placeholder="Search..." />
      </div>
      <div class="ms-auto d-flex align-items-center">
        <div class="dropdown">
          <a
            class="nav-link dropdown-toggle"
            href="#"
            role="button"
            data-bs-toggle="dropdown">
            <i class="bi bi-person-circle fs-5"></i>
          </a>
          <ul class="dropdown-menu dropdown-menu-end">
            <li>
              <a class="dropdown-item" href="profile.php">Profile</a>
            </li>
            <li>
              <a class="dropdown-item" href="settings.html">Settings</a>
            </li>
            <li>
              <hr class="dropdown-divider" />
            </li>
            <li><a class="dropdown-item" href="logout.php">Logout</a></li>
          </ul>
        </div>
      </div>
    </div>
  </nav>

  <div class="appointments-container">
    <div class="row">
      <div class="col-md-8">
        <div class="card">
          <div class="card-body">
            <h4 class="card-title mb-4">Schedule Appointment</h4>
            <form action="appointments.php" method="POST" id="appointmentForm">
              <div class="mb-3">
                <label class="form-label">Doctor/Physician</label>
                <select class="form-select" name="doctor_name" required>
                  <option value="Dr. Sarah Johnson - Cardiologist">Dr. Sarah Johnson - Cardiologist</option>
                  <option value="Dr. Michael Chen - General Physician">Dr. Michael Chen - General Physician</option>
                  <option value="Dr. Emily Brown - Neurologist">Dr. Emily Brown - Neurologist</option>
                </select>
              </div>
              <div class="mb-3">
                <label class="form-label">Date</label>
                <input type="date" class="form-control" name="appointment_date" required />
              </div>
              <div class="mb-3">
                <label class="form-label">Time Slot</label>
                <select class="form-select" name="appointment_time" required>
                  <option value="09:00 AM">09:00 AM</option>
                  <option value="10:00 AM">10:00 AM</option>
                  <option value="11:00 AM">11:00 AM</option>
                  <option value="02:00 PM">02:00 PM</option>
                  <option value="03:00 PM">03:00 PM</option>
                </select>
              </div>
              <div class="mb-3">
                <label class="form-label">Reason for Visit</label>
                <textarea class="form-control" rows="3" name="notes" required></textarea>
              </div>
              <input type="hidden" name="user_id" value="<?php echo $_SESSION['user_id']; ?>" /> <!-- Add hidden user_id -->
              <button type="submit" class="btn btn-primary">Schedule Appointment</button>
            </form>

          </div>
        </div>
      </div>
      <div class="col-md-4">
        <div class="card">
```

```
       <div class="card-body">
        <h4 class="card-title mb-4">Upcoming Appointments</h4>
        <div class="upcoming-appointments">
         <?php
         // Fetch upcoming appointments for the logged-in user
         $user_id = $_SESSION['user_id']; // Get the user ID from the session
         $sql_upcoming = "SELECT * FROM appointments WHERE user_id = ? AND appointment_date >= CURDATE() ORDER BY
appointment_date ASC";
         $stmt = $conn->prepare($sql_upcoming);
         $stmt->bind_param("i", $user_id);
         $stmt->execute();
         $result = $stmt->get_result();

         // Display upcoming appointments
         while ($row = $result->fetch_assoc()) {
          $appointment_date = date('d M', strtotime($row['appointment_date']));
          $doctor_name = $row['doctor_name'];
          $appointment_time = $row['appointment_time'];
          echo "<div class='appointment-item'>";
          echo "<div class='appointment-date'><span class='day'>" . date('d', strtotime($row['appointment_date'])) .
"</span><span class='month'>" . date('M', strtotime($row['appointment_date'])) . "</span></div>";
          echo "<div class='appointment-details'><h6>$doctor_name</h6><p>$appointment_time</p></div>";
          echo "</div>";
         }

         $stmt->close();
         ?>
        </div>
       </div>
      </div>
     </div>
    </div>
   </div>
  </div>
 </div>

 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>
 <script src="assets/js/appointments.js"></script>
</body>

</html>
```

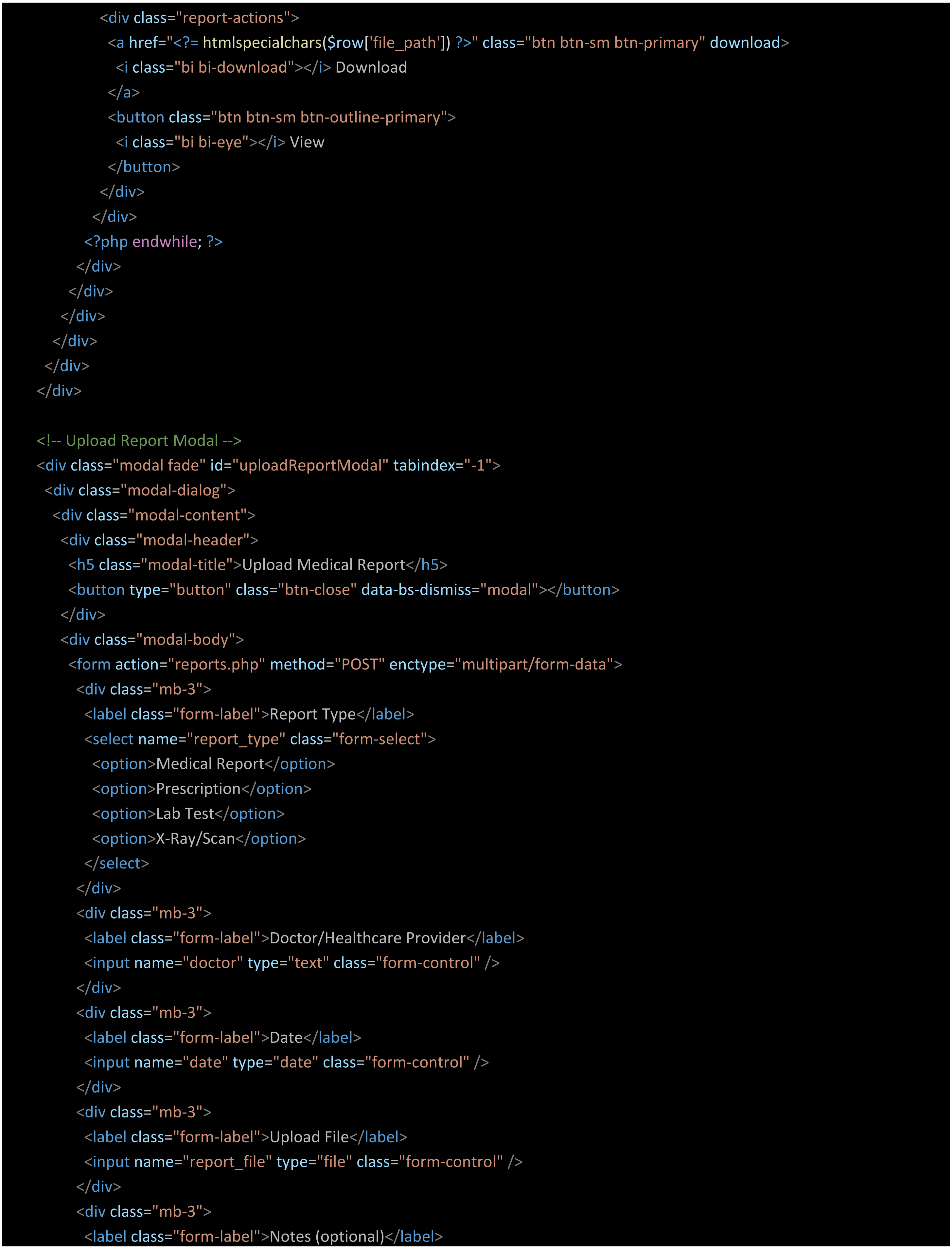
Reports.php

```
<?php
// Start the session to manage user data
session_start();

include 'db.php';
```

```php
include 'create_tables.php'; // Include the table creation script

// If the user is not logged in, redirect them to the login page
if (!isset($_SESSION['user_id'])) {
    header("Location: login.php");
    exit();
}

// Get the user_id from the session
$user_id = $_SESSION['user_id'];

// Handling file upload
if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_FILES['report_file'])) {
    $report_type = $_POST['report_type'];
    $doctor = $_POST['doctor'];
    $date = $_POST['date'];
    $notes = $_POST['notes'];

    $file_name = $_FILES['report_file']['name'];
    $file_tmp_name = $_FILES['report_file']['tmp_name'];
    $file_size = $_FILES['report_file']['size'];
    $file_error = $_FILES['report_file']['error'];

    // Validate file upload
    if ($file_error === 0) {
        // Set a target path for the file
        $target_dir = "uploads/reports/";
        $target_file = $target_dir . basename($file_name);

        // Move the file to the server directory
        if (move_uploaded_file($file_tmp_name, $target_file)) {
            // Insert data into the database, including the user_id from the session
            $stmt = $conn->prepare("INSERT INTO reports (user_id, file_type, doctor, date, file_path, notes) VALUES (?, ?, ?, ?, ?, ?)");
            $stmt->bind_param("isssss", $user_id, $report_type, $doctor, $date, $target_file, $notes);
            $stmt->execute();
            $stmt->close();
        } else {
            echo "Error uploading file.";
        }
    } else {
        echo "Error with file upload.";
    }
}

// Fetch reports from the database
$sql = "SELECT * FROM reports ORDER BY upload_date DESC";
$result = $conn->query($sql);
?>
```



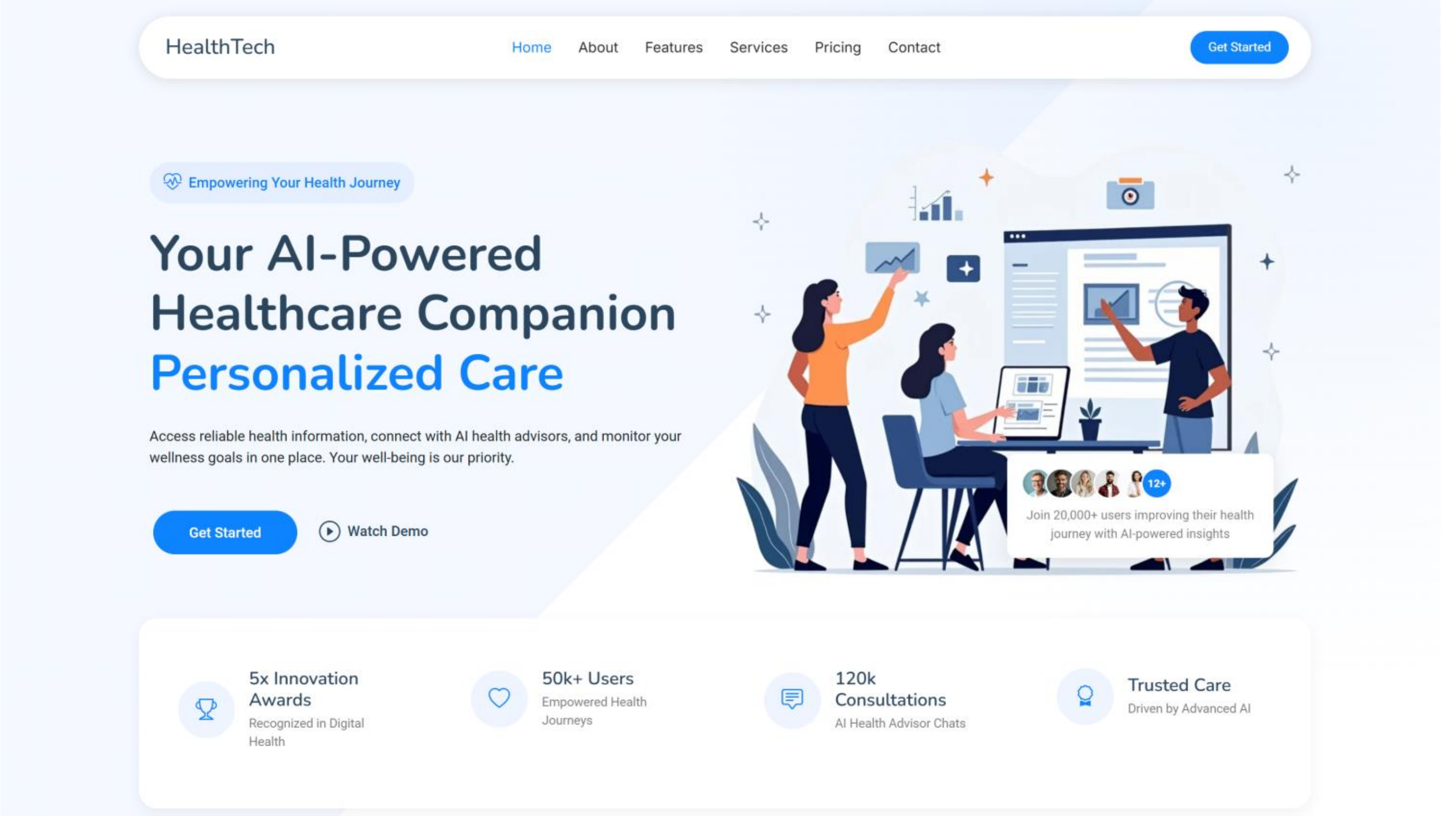
```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Medical Reports - AI Dashboard</title>
    <link
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
      rel="stylesheet"
    />
    <link
      href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.1/font/bootstrap-icons.css"
      rel="stylesheet"
    />
    <link href="assets/css/styles.css" rel="stylesheet" />
    <link href="assets/css/reports.css" rel="stylesheet" />
  </head>
  <body>
    <div class="wrapper">
      <nav id="sidebar" class="active">
        <div class="sidebar-header">
          <h3>AI Dashboard</h3>
        </div>
        <ul class="list-unstyled components">
          <li class="active">
            <a href="dashboard.php"><i class="bi bi-chat-dots-fill me-2"></i>Chat</a>
          </li>
          <li>
            <a href="dashboard.php#features"><i class="bi bi-grid-fill me-2"></i>Features</a>
          </li>
          <li>
            <a href="profile.php"><i class="bi bi-person-fill me-2"></i>Profile</a>
          </li>
          <li>
            <a href="settings.php"><i class="bi bi-gear-fill me-2"></i>Settings</a>
          </li>
          <li>
            <a href="appointments.php"><i class="bi bi-calendar-check me-2"></i>Appointments</a>
          </li>
          <li>
            <a href="reports.php"><i class="bi bi-file-medical me-2"></i>Reports</a>
          </li>
          <li>
            <a href="goals.php"><i class="bi bi-trophy me-2"></i>Health Goals</a>
          </li>
        </ul>
      </nav>

      <div id="content">
        <nav class="navbar navbar-expand-lg navbar-light bg-light">
          <div class="container-fluid">
            <button type="button" id="sidebarCollapse" class="btn btn-primary">
              <i class="bi bi-list"></i>
            </button>
            <div class="ms-2">
              <input type="text" class="form-control" placeholder="Search reports..." />
            </div>
            <div class="ms-auto d-flex align-items-center gap-2">
              <button class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#uploadReportModal">
                <i class="bi bi-upload me-2"></i>Upload Report
              </button>
              <div class="dropdown">
                <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown">
                  <i class="bi bi-person-circle fs-5"></i>
                </a>
                <ul class="dropdown-menu dropdown-menu-end">
                  <li><a class="dropdown-item" href="profile.php">Profile</a></li>
                  <li><a class="dropdown-item" href="settings.php">Settings</a></li>
                  <li><hr class="dropdown-divider" /></li>
                  <li><a class="dropdown-item" href="logout.php">Logout</a></li>
                </ul>
              </div>
            </div>
          </div>
        </nav>

        <div class="reports-container">
          <div class="row">
            <div class="col-md-12">
              <div class="card">
                <div class="card-body">
                  <div class="d-flex justify-content-between align-items-center mb-4">
                    <h4 class="card-title mb-0">Medical Reports & Prescriptions</h4>
                    <div class="btn-group">
                      <button class="btn btn-outline-primary active">All</button>
                      <button class="btn btn-outline-primary">Reports</button>
                      <button class="btn btn-outline-primary">Prescriptions</button>
                    </div>
                  </div>
                  <div class="reports-list">
                    <?php while ($row = $result->fetch_assoc()): ?>
                      <div class="report-item">
                        <div class="report-icon">
                          <i class="bi bi-file-earmark-medical"></i>
                        </div>
                        <div class="report-details">
                          <h5><?= htmlspecialchars($row['file_type']) ?></h5>
                          <p><?= htmlspecialchars($row['doctor']) ?></p>
                          <small><?= htmlspecialchars($row['date']) ?></small>
                        </div>
```



```
        <div class="report-actions">
         <a href="<?= htmlspecialchars($row['file_path']) ?>" class="btn btn-sm btn-primary" download>
          <i class="bi bi-download"></i> Download
         </a>
         <button class="btn btn-sm btn-outline-primary">
          <i class="bi bi-eye"></i> View
         </button>
        </div>
       </div>
      <?php endwhile; ?>
     </div>
    </div>
   </div>
  </div>
 </div>
</div>

<!-- Upload Report Modal -->
<div class="modal fade" id="uploadReportModal" tabindex="-1">
 <div class="modal-dialog">
  <div class="modal-content">
   <div class="modal-header">
    <h5 class="modal-title">Upload Medical Report</h5>
    <button type="button" class="btn-close" data-bs-dismiss="modal"></button>
   </div>
   <div class="modal-body">
    <form action="reports.php" method="POST" enctype="multipart/form-data">
     <div class="mb-3">
      <label class="form-label">Report Type</label>
      <select name="report_type" class="form-select">
       <option>Medical Report</option>
       <option>Prescription</option>
       <option>Lab Test</option>
       <option>X-Ray/Scan</option>
      </select>
     </div>
     <div class="mb-3">
      <label class="form-label">Doctor/Healthcare Provider</label>
      <input name="doctor" type="text" class="form-control" />
     </div>
     <div class="mb-3">
      <label class="form-label">Date</label>
      <input name="date" type="date" class="form-control" />
     </div>
     <div class="mb-3">
      <label class="form-label">Upload File</label>
      <input name="report_file" type="file" class="form-control" />
     </div>
     <div class="mb-3">
      <label class="form-label">Notes (optional)</label>
```

```
                    <textarea name="notes" class="form-control"></textarea>
                  </div>
                  <button type="submit" class="btn btn-primary w-100">Upload Report</button>
                </form>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>
    <script src="assets/js/reports.js"></script>
  </body>
</html>
```

7. Screenshots



Your AI-Powered Healthcare Companion Personalized Care

Access reliable health information, connect with AI health advisors, and monitor your wellness goals in one place. Your well-being is our priority.

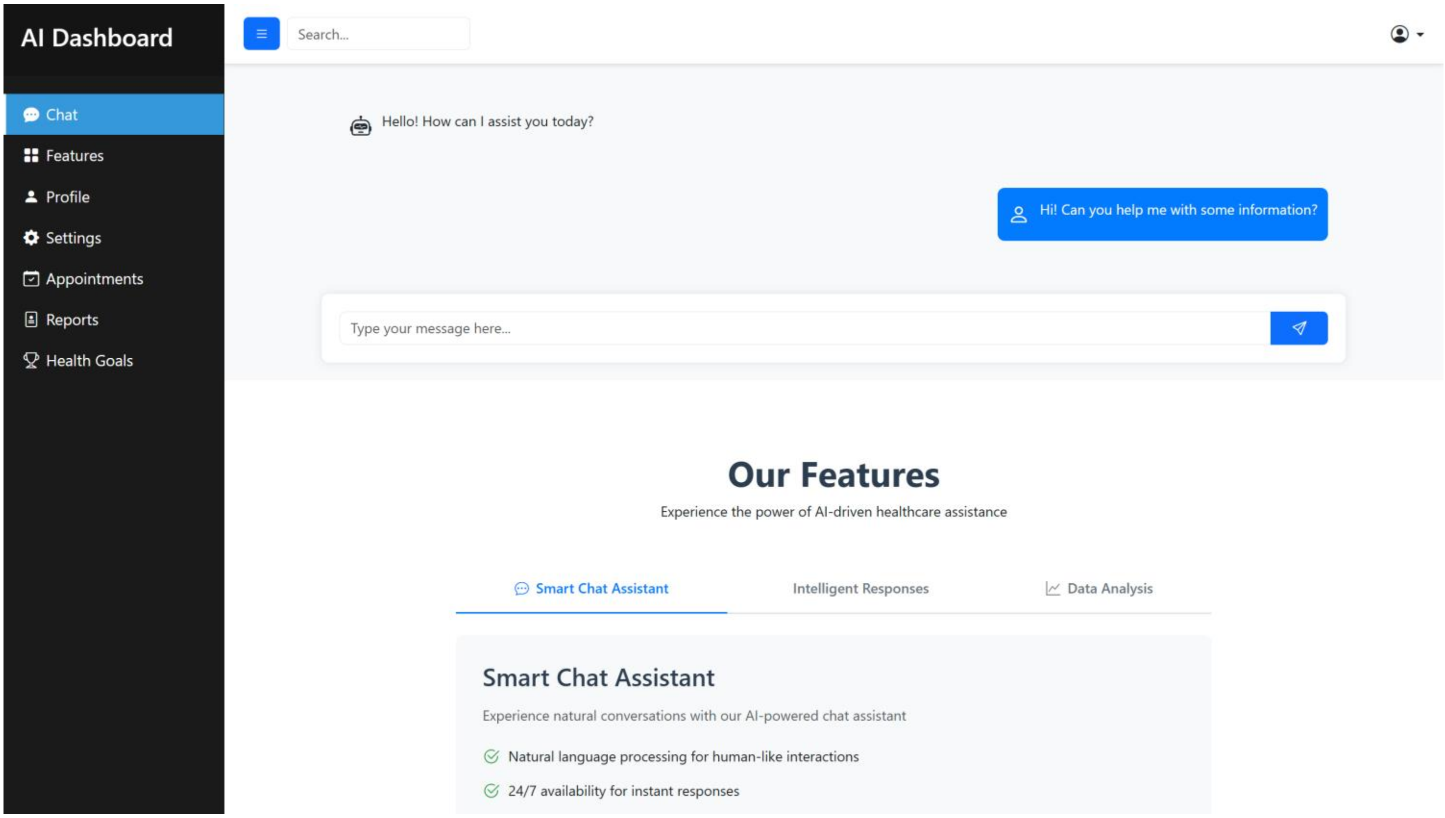
Join 20,000+ users improving their health journey with AI-powered insights

5x Innovation Awards
Recognized in Digital Health

50k+ Users
Empowered Health Journeys

120k Consultations
AI Health Advisor Chats

Trusted Care
Driven by Advanced AI



Hello! How can I assist you today?

Hi! Can you help me with some information?

Type your message here...

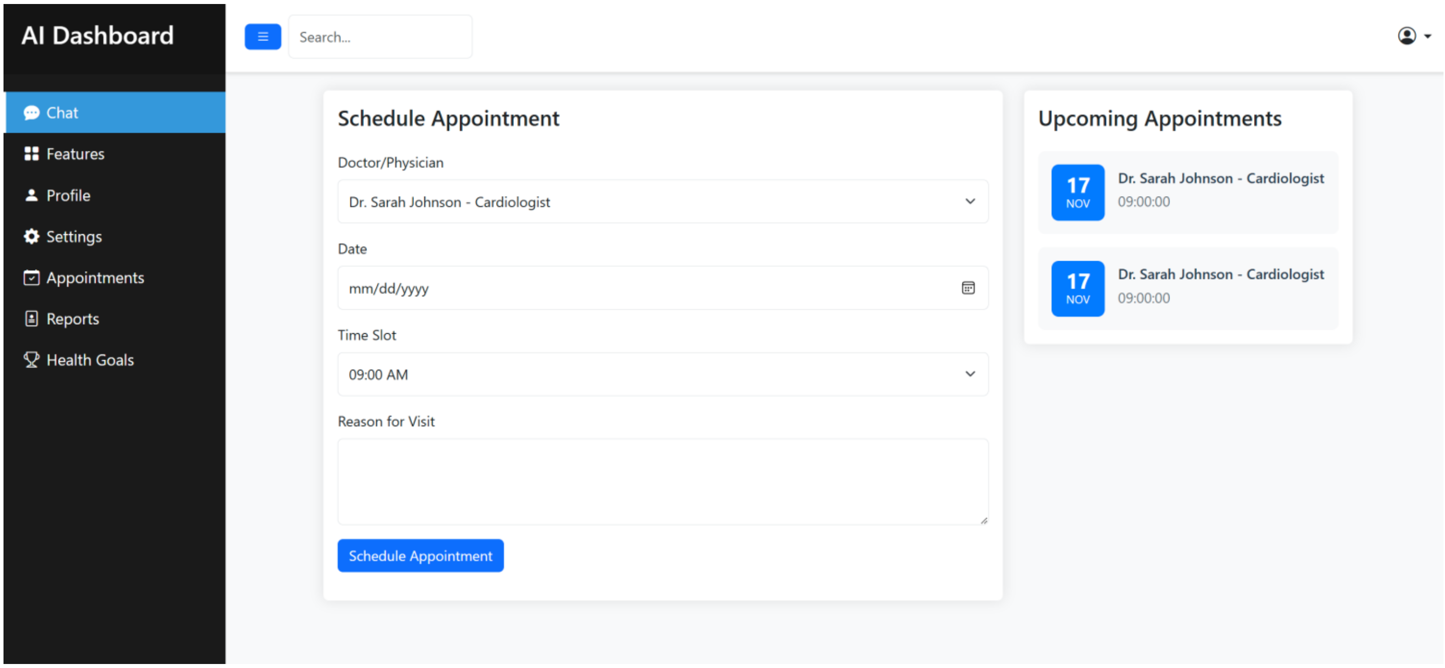
Our Features

Experience the power of AI-driven healthcare assistance

Smart Chat Assistant

Experience natural conversations with our AI-powered chat assistant

- ✓ Natural language processing for human-like interactions
- ✓ 24/7 availability for instant responses



Schedule Appointment

Doctor/Physician
Dr. Sarah Johnson - Cardiologist

Date
mm/dd/yyyy

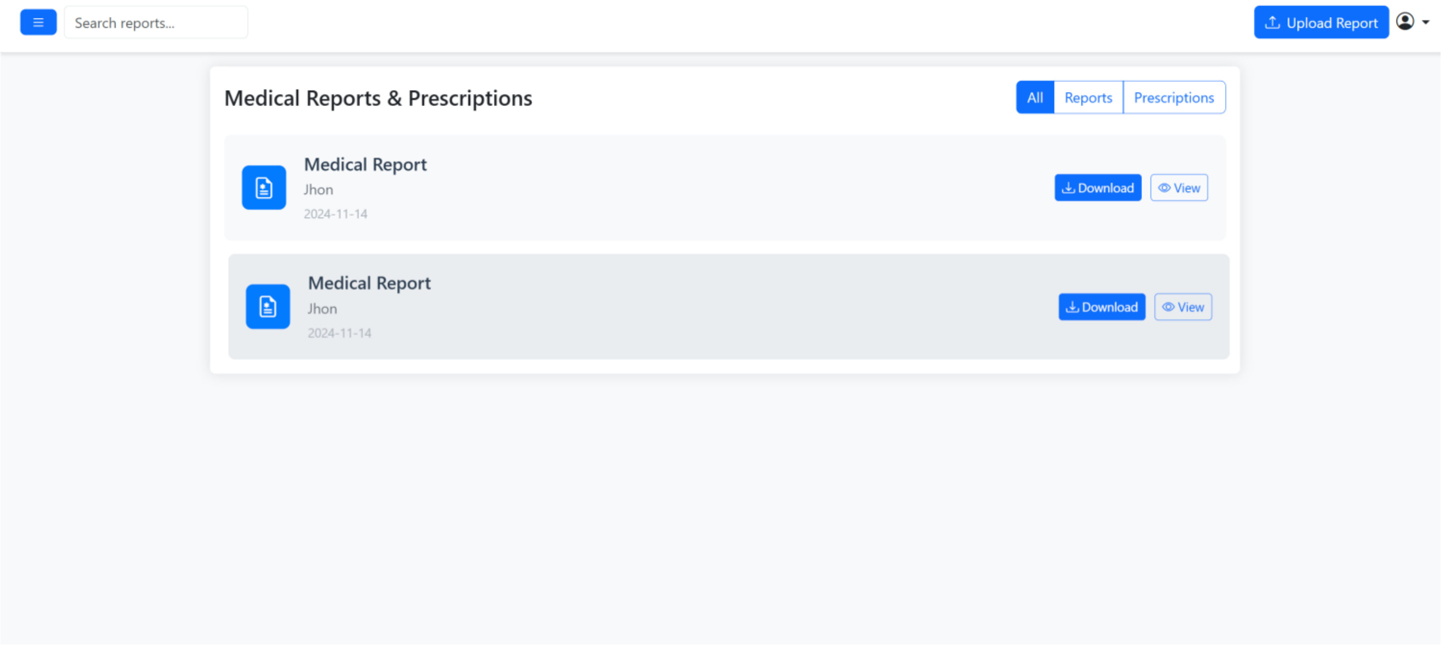
Time Slot
09:00 AM

Reason for Visit

Upcoming Appointments

17 NOV Dr. Sarah Johnson - Cardiologist
09:00:00

17 NOV Dr. Sarah Johnson - Cardiologist
09:00:00



Medical Reports & Prescriptions

Medical Report
Jhon
2024-11-14

Medical Report
Jhon
2024-11-14

8. Conclusion

8.1. Summary of the Project

The **AI Healthcare Assistant** project aimed to develop an intelligent web application that assists users in managing their healthcare by integrating several essential features such as AI-powered chat assistance, health goal tracking, appointment management, and the ability to upload and manage medical reports. The project utilized **PHP** for backend development, **MySQL** for database management, and an AI-based chatbot to provide personalized healthcare advice. The system creates and updates a personalized reference file for each user, which serves as a comprehensive medical history, and helps provide relevant suggestions during future interactions.

By implementing features such as secure user authentication, appointment scheduling, health goal management, and a chatbot capable of interpreting and responding to user queries, this project aims to enhance healthcare access and patient engagement. Users can now track and manage their health data more effectively while receiving AI-driven support.

8.2. Achievements

Several key achievements were accomplished throughout the development of this project:

- **AI Chatbot Integration**: Successfully developed and integrated an AI-powered chatbot capable of assisting users with health-related queries and providing personalized recommendations based on user data.
- **Healthcare Management Features**: Enabled the uploading and managing of medical reports, prescriptions, and health goals, allowing users to keep their health data organized and accessible.
- **User Authentication and Data Security**: Implemented secure user authentication and session management using PHP, ensuring that user data is protected and private.
- **Appointment Scheduling**: Developed a robust appointment scheduling system that allows users to book, manage, and track their medical appointments.
- **Personalized Health Tracking**: Created a dynamic reference file for each user, which updates automatically with new interactions, such as medical reports, prescriptions, and health goal progress, providing context-sensitive advice during future interactions.

These achievements demonstrate the potential of integrating AI and web technologies into the healthcare sector, improving user experience, accessibility, and data management.

8.3. Challenges and Lessons Learned

During the development of this project, several challenges were encountered:

- **Integration of AI**: One of the significant challenges was ensuring that the AI chatbot provided relevant and accurate health-related advice. This required careful design of the chatbot's capabilities and limitations to ensure that it could offer helpful responses while avoiding medical overreach.
- **Data Privacy and Security**: Protecting sensitive user data posed another challenge, particularly with healthcare data. Implementing secure authentication, encryption, and ensuring that data was properly sanitized before being stored in the database were critical aspects of the project.
- **Managing Complex Interactions**: The reference file feature required careful handling of data updates and ensuring that multiple data types (reports, prescriptions, appointments, and goals) could be managed in a cohesive manner.
- **System Performance**: Ensuring that the application performed optimally, especially with the growing volume of user data, was crucial. Efforts to optimize database queries and improve the efficiency of the AI chatbot were necessary to maintain smooth user interaction.

From these challenges, the key lesson learned was the importance of thoroughly planning the architecture of a web-based application to ensure scalability and security. Additionally, balancing the complexity of AI-driven interactions with user-friendly interfaces was essential to maintaining a positive user experience.

---

9. References

9.1. Books and Articles

1. Welling, L., & Thomson, L. (2016). *PHP and MySQL Web Development* (5th ed.). Addison-Wesley.
2. Kolb, C. (2020). *Practical Machine Learning for Healthcare* (1st ed.). O'Reilly Media.
3. Gupta, A. (2019). *Artificial Intelligence in Healthcare: Past, Present, and Future*. Springer International Publishing.
4. Ahmed, Z. (2021). *Building Chatbots with PHP and MySQL*. Packt Publishing.

5. Rouse, M. (2018). *Healthcare Analytics: From Data to Decisions*. Springer.

9.2. Websites and Documentation

1. PHP Official Documentation: https://www.php.net/docs.php
2. MySQL Documentation: https://dev.mysql.com/doc/
3. Bootstrap Documentation: https://getbootstrap.com/docs/5.0/getting-started/introduction/
4. OpenAI GPT-3 Documentation: https://beta.openai.com/docs/
5. PHP Manual for Sessions: https://www.php.net/manual/en/book.session.php

---

10. Appendices

10.1. Appendix A: Code Listings

- Code for **User Authentication**: Includes session handling and secure login functionality.
- Code for **AI Chatbot**: Code that powers the AI assistant, including the integration with the reference file system.
- Code for **Appointment Scheduling System**: Backend logic for managing appointments.

10.2. Appendix B: Database Schema

- **Users Table**: Holds user information such as names, emails, and passwords.
- **Appointments Table**: Stores appointment details for each user.
- **Health Goals Table**: Tracks health-related goals, progress, and completion.
- **Reports Table**: Stores uploaded reports and prescriptions linked to each user.
- **Chat Logs Table**: Logs the interactions between users and the AI chatbot, which are used for reference in future conversations.
-

11. Github and Hosted Link

11.1. Github Repository Link

You can access the complete source code of the project on the following GitHub repository:

Github Repository Link : hvsk004/AI-HealthTech

11.2. Live Website URL

The live version of the AI Healthcare Assistant website can be accessed at:

Live Website URL : hvsaikrishna.dev/AI-HealthTech