**Egyptian E-Learning University**

Faculty of Computers & Information Technology

Hospital Management System with Deep Learning

**By**

|  |  |
| --- | --- |
| Hader Mahmoud Shaban | 21-02137 |
| Mariam Medhat Naeem | 21-02083 |
| Merna Saeed Habib | 21-02108 |
| Demiana Aiad Eshak | 21-01896 |
| Rana Ragab Hassan | 21-01904 |
| Ahmed Mohamed Ahmed | 21-01841 |
| Mohamed Sayed Gomaa | 21-02041 |
| Mahmoud Essam Abdel Star | 21-02066 |

Supervised by

Dr. **Enas Farouk**

Assistant

Eng. **Rehab Galal**

[Beni-suef]-2025

Abstract

This project introduces a comprehensive Hospital Management System developed using [ASP.NET](https://asp.net/) Web Forms for the backend and SQL Server for the database.

The main goal was to create an integrated platform that manages hospital operations efficiently and supports medical diagnosis through machine learning. The system handles core hospital functions such as patient registration, appointment scheduling, staff and doctor management, billing, and treatment history.

In addition, the project features a unique MRI image-based prediction module that allows users to upload brain or skin MRI scans. These are processed by a pre-trained machine learning model to detect the presence of tumors (such as glioma, meningioma, pituitary tumors, or skin cancer).

The backend logic was built using [ASP.NET](https://asp.net/) Web Forms, enabling dynamic interaction between the user interface and the server.

The machine learning model was developed and integrated using Python and TensorFlow, and the image upload functionality was smoothly connected to the prediction API.

Testing with real users (students simulating hospital roles) confirmed the system’s ease of use and accuracy in diagnosis. The interface was appreciated for its clarity, and minor feedback led to improved error messages and labels.

In summary, the system successfully combines hospital management with modern AI-based diagnostic tools, providing both administrative efficiency and medical intelligence.

Acknowledgments

I would like to express my deepest gratitude to all those who have supported and guided me throughout the development of this hospital management system project.

First and foremost, I am profoundly grateful to (Eng. Rehab Galal) for her invaluable technical guidance and continuous encouragement. Her expertise and insights were instrumental in shaping the direction and success of this project.

I would also like to extend my heartfelt thanks to (Dr. Enas Farouk) for her academic support and constructive feedback. Her dedication and commitment to excellence have been a source of inspiration throughout this journey.

Finally, I am thankful to my family and friends for their unwavering support and understanding during this challenging yet rewarding endeavor.

Contents

[Abstract 2](#_Toc197463970)

[Acknowledgments 3](#_Toc197463971)

[Introduction 5](#_Toc197463972)

[Literature Review / Related Work 10](#_Toc197463973)

[Proposed system 14](#_Toc197463974)

[Implementation 21](#_Toc197463975)

[Testing & Evaluation 24](#_Toc197463976)

[Results & Discussion 27](#_Toc197463977)

[Conclusion & Future Work 29](#_Toc197463978)

[References 31](#_Toc197463979)

[Appendices (Optional) 34s](#_Toc197463980)

Chapter 1

Introduction

* 1. **Introduction**

This project is a web-based Hospital Management System developed using [ASP.NET](https://asp.net/) Web Forms and SQL Server. It includes essential hospital operations such as patient records, appointments, billing, and staff management. In addition, it supports a machine learning module that allows doctors to upload MRI images and receive predictions about brain or skin tumors using artificial intelligence.

In the modern healthcare environment, the efficient management of hospital operations is crucial for delivering high-quality patient care.

The Hospital Management System (HMS) project aims to streamline and automate various administrative and clinical processes within a hospital.

This system is designed to manage patient information, appointments, billing, and medical records, ensuring that all data is securely stored and easily accessible to authorized personnel.

The HMS will enhance the overall efficiency of hospital operations by reducing manual tasks, minimizing errors, and improving communication among healthcare providers. By leveraging advanced technologies, the system will provide a user-friendly interface for both medical staff and patients, facilitating better healthcare delivery and patient satisfaction.

This documentation outlines the development process, system design, implementation strategies, and testing methodologies employed in the creation of the Hospital Management System.

* 1. **Background and motivation for the project**

Hospitals face many challenges managing daily operations manually, This approach can lead to several issues (errors, delays, and poor patient experiences) including:

1. **Inefficiency**: Manual processes are time-consuming and prone to human error, which can delay patient care and administrative tasks.
2. **Data Inaccuracy**: Paper-based records and fragmented digital systems can result in inconsistent and inaccurate patient information, leading to potential medical errors.
3. **Poor Communication**: Lack of integration between departments can hinder effective communication among healthcare providers, affecting the quality of patient care.
4. **Resource Management**: Inefficient scheduling and resource allocation can lead to underutilization or overburdening of hospital resources, impacting overall operational efficiency.
5. **Patient Experience**: Long wait times, difficulty in accessing medical records, and lack of transparency in billing can negatively affect patient satisfaction.

-At the same time, early detection of tumors is critical in saving lives, yet not always easily accessible. We wanted to build a system that improves both hospital management and medical diagnosis using modern technologies..

* 1. **Importance of the problem being addressed**

-Many hospitals still use paper-based systems or outdated software.

-Manual work causes delays in scheduling, billing, and finding records.

-Tumor detection often requires expert analysis, which may not be available in all hospitals.

-Our system solves both problems by: Automating hospital tasks Adding a machine learning model to assist doctors with early diagnosis

* 1. **Problem Statement**

-Many hospitals lack a unified digital system that handles both management tasks and smart medical tools. This can lead to errors, lost time, and late diagnosis.

-Why this matters:

Without proper systems, patients may wait longer, records can be lost, and doctors may not have easy access to smart tools for tumor detection. These issues affect patient safety and hospital efficiency.

* 1. **Objectives**

-Main Objective:

To develop a complete Hospital Management System that also integrates tumor prediction using MRI images.

-Specific Objectives:

1-Create a database to manage patients, doctors, and appointments

2-Design an easy-to-use user interface

3-Implement secure login for different user roles

4-Allow image upload and run prediction using a trained ML model

5-Generate results with prediction confidence

6-Test the system with real users and improve based on feedback

7-Billing Automation

8-Diagnosis Recording

* 1. **Brief overview of the proposed solution**.

-Our solution is a web-based platform that helps hospitals manage their operations digitally.

-It allows admins, doctors, and patients to log in, view information, and perform tasks like scheduling, billing, and treatment tracking.

-It also includes a smart feature where patients can upload an MRI image, and the system will predict the type of tumor (or no tumor) using a deep learning model with high accuracy.

Chapter 2

Literature Review / Related Work

**2.1** **Summary of existing research and technologies related to your project.**

The current manual system has a lot of paperwork.

To maintain the records of sale and service manually, is a Time-consuming task.

With the increase in databases, it will become a massive task to maintain the database.

Requires large quantities of file cabinets, which are huge and require quite a bit of space in the office, which can be used for storing records of previous details.

The retrieval of records of previously registered patients will be a tedious task.

Lack of security for the records, no one disarranges the records of your system.

If someone wants to check the details of the available doctors the previous system does not provide any necessary details of this type.

All this work is done manually by the receptionist and other operational staff and lot of papers are needed to be handled and taken care of.

Doctors must remember various medicines available for diagnosis and sometimes miss better alternatives as they can't remember them at that time.

-overall Problems with Existing system:

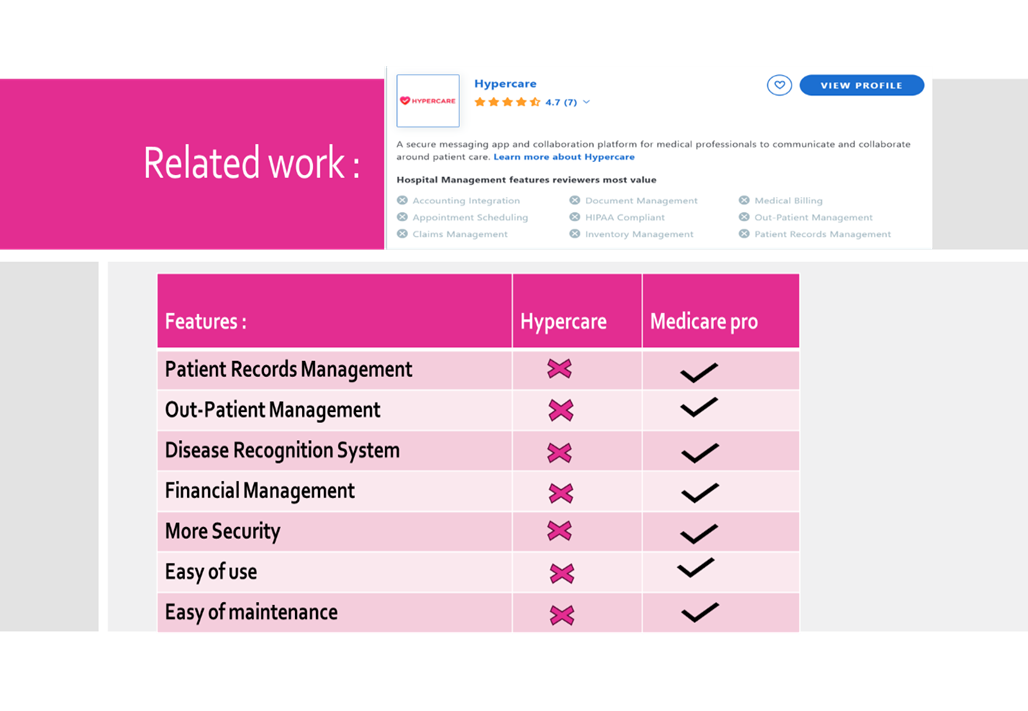
1. Inefficiency in Administrative Processes
2. Inaccurate and Fragmented Data
3. Coordination
4. Resource Management Challenges:
5. Long Wait Times and Poor Patient Experience
6. Security and Privacy Concerns
7. Compliance with Healthcare Regulations
8. Dependence on Outdated Systems
9. High Initial Costs and Implementation Time
10. User Adoption and Training

**2.2 Gaps in current solutions that your project aims to fill.**

-Most hospital management systems do not include AI-based diagnostic features. -Tumor detection models are typically built for research purposes and are not integrated into practical hospital systems.

-Medical professionals are often forced to use multiple platforms for administration and diagnostics, leading to inefficiency.

-Many current systems have outdated interfaces or lack user-friendly design and real-time features.



**2.3 Summary**

While separate solutions exist for hospital management and machine learning-based medical diagnostics, there is a clear lack of platforms that unify both.

Our project bridges this gap by offering a system that not only manages hospital workflows efficiently but also supports doctors in early diagnosis through intelligent tumor prediction from MRI images.

This combination aims to enhance both operational efficiency and healthcare quality.

Chapter 3

Proposed system

* 1. **Approach used to solve the problem**

The proposed system aims to integrate hospital management with intelligent tumor detection using AI.

The main idea is to offer a web-based platform that automates hospital processes (like patient registration, appointments, staff management) and allows doctors to upload MRI or skin images to detect possible tumors using trained machine learning models.

We chose a modular approach:

1- [ASP.NET](https://asp.net/) Web Forms for the backend and frontend.

2-A SQL Server database for managing patient records, doctor schedules, and hospital data.

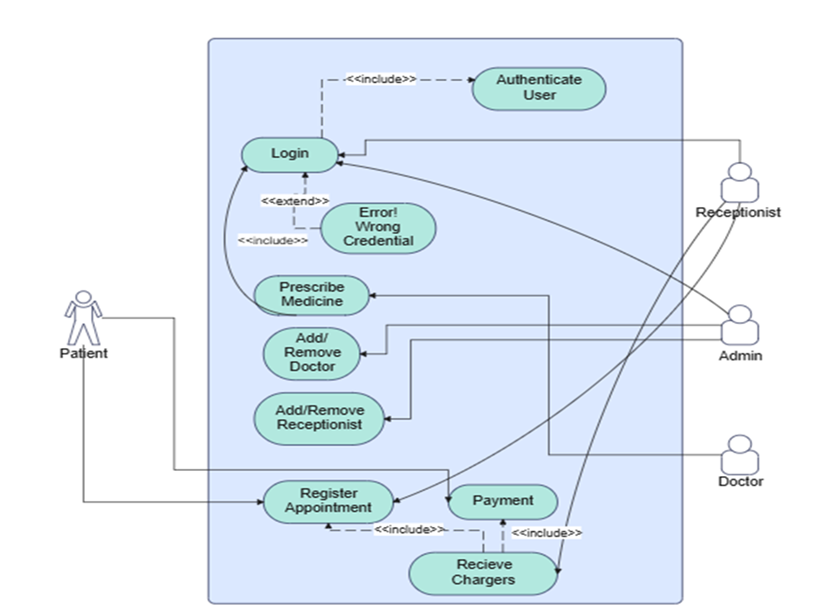
3-Trained machine learning models (for brain and skin tumor detection) that were developed using Python (Kaggle - Jupyter), then saved and uploaded for prediction through a web interface.

This approach ensures that hospital staff can handle daily tasks while also improving medical diagnostics with AI support**.**

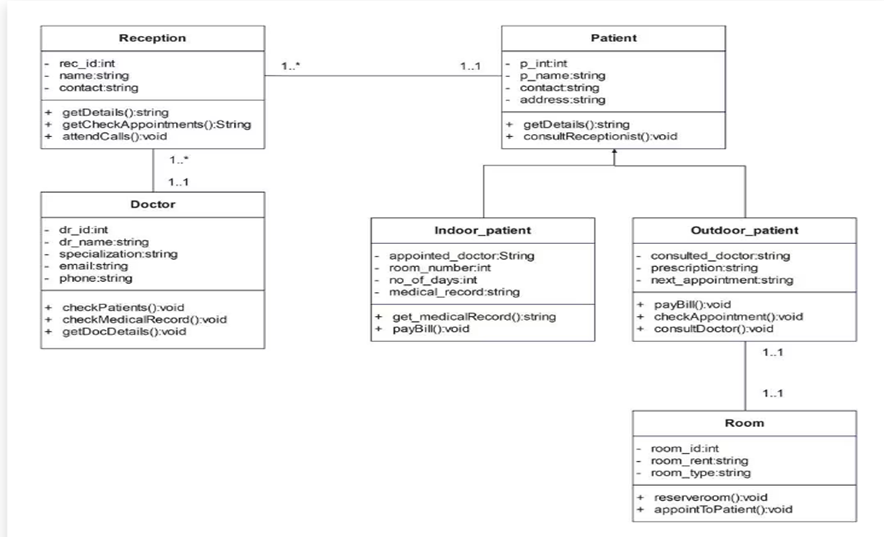
* 1. **System architecture**

**UML**

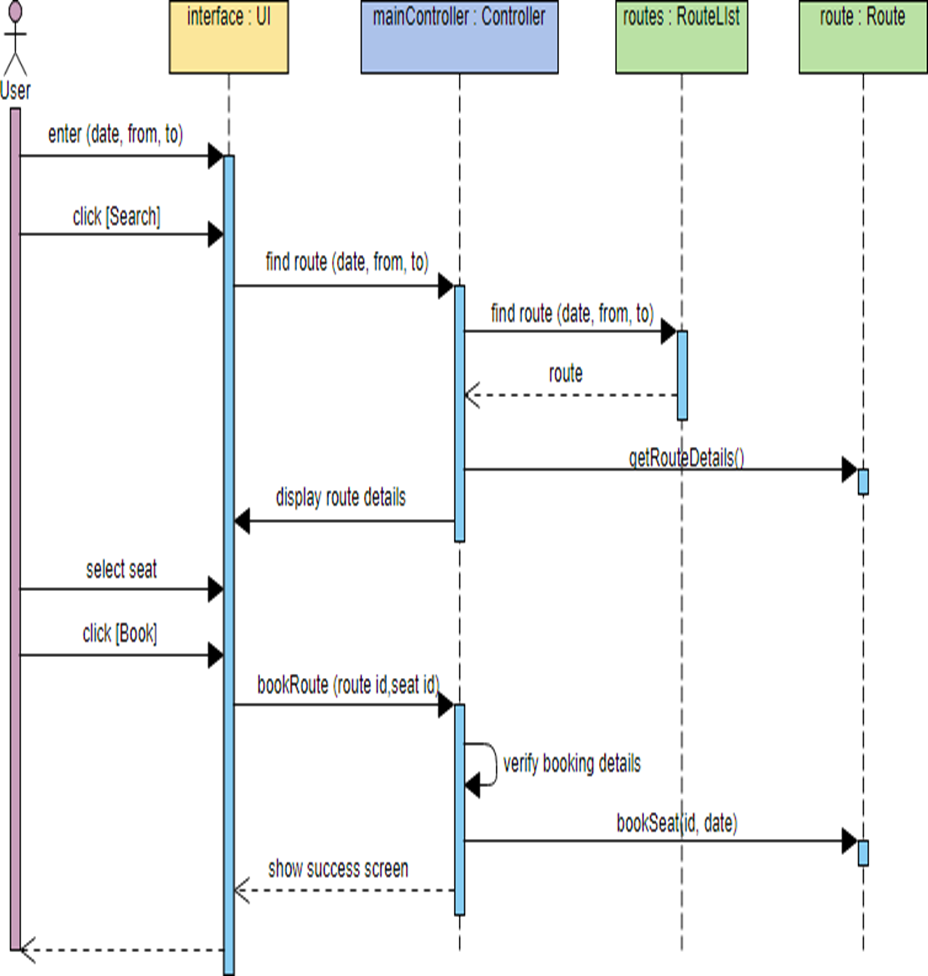
1-UML Use Case Diagram



2-UML Activity Diagram



3- UML Sequence Diagram

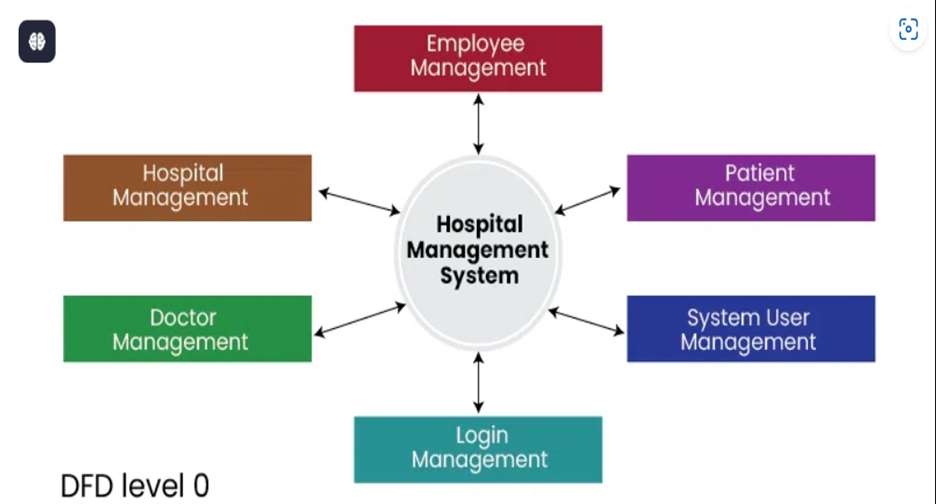


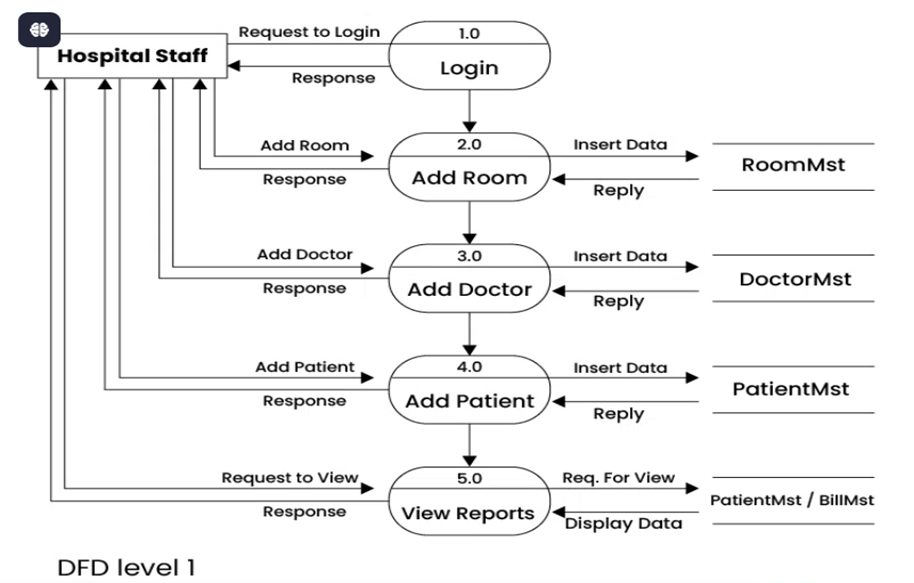
4- UML Class Diagram

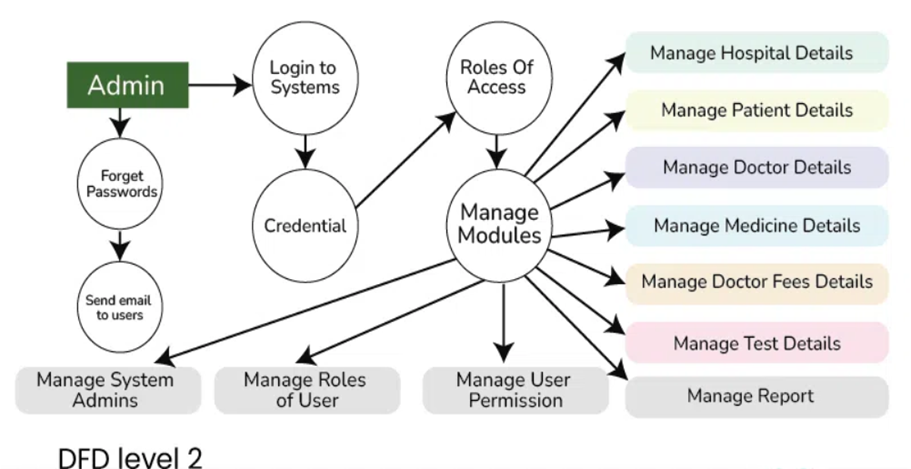
A diagram of a medical organization

AI-generated content may be incorrect.

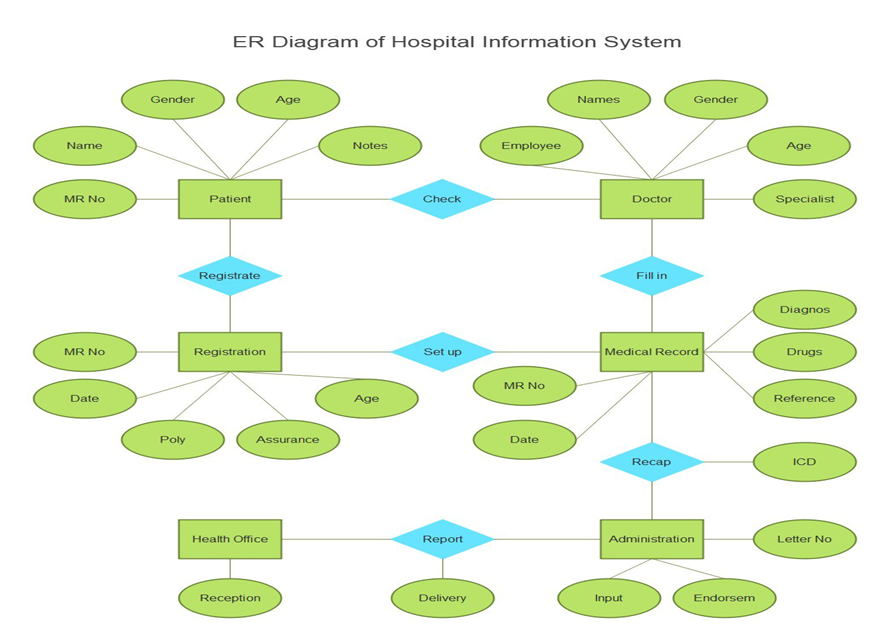
**Flowcharts**

****

****

****

**ER diagrams**

****

* 1. **Algorithms or frameworks used.**

1-[ASP.NET](https://asp.net/) Web Forms – Used for building dynamic web pages and form submissions.

2-C# – Logic for user roles, validations, and form submissions.

3-SQL Server – For storing user details, appointments, and medical records.

4-Tumor Detection:

- Brain Tumor: A Custom CNN model trained on MRI datasets (e.g., from Kaggle) to detect glioma, meningioma, and pituitary tumors.

- Skin Cancer: A CNN model trained to classify benign vs malignant skin lesions. 5-Python Libraries: TensorFlow and Keras for training and saving models.

sklearn for evaluation metrics.

Chapter 4

Implementation

* 1. **Technologies, tools, and programming languages used**

For the development of the hospital management system, we used the following technologies:

-Backend: [ASP.NET](https://asp.net/) with Web Forms for handling server-side logic and database interaction.

-Frontend: HTML, CSS, and JavaScript to build the user interface.

-Database: Microsoft SQL Server for storing patients, doctors, appointments, and billing data.

Tools:

-Py-cache ,Jupyter Notebook and Kaggle were used to train and test deep learning models for skin cancer and brain tumor detection using MRI images.

-Visual Studio (for ASP.NET and C#)

Model Integration: Trained models were exported and integrated into the [ASP.NET](https://asp.net/) system to provide predictions during image uploads.

Programming languages:

-C# (for ASP.NET)

-Python (for machine learning and deep learning)

* 1. **Key components/modules of the system**.

-User Management: Handles login/signup for doctors, patients, and admins.

-Appointment Scheduling: Patients can book appointments with doctors and select the time and date and their department and view upcoming visits.

-Medical Image Prediction: Patients can upload brain MRI or skin images to get predictions about tumors.

-Billing & Reports: Admins and patients can view bills and treatment history.

-Feedback System: Patients can give feedback after visits.

-Notification page for patient: to view important updates related to their health, appointments, test results, and system alerts.

* 1. **Challenges faced and how they were resolved**

1-Model Deployment: Integrating Python-trained models into an [ASP.NET](https://asp.net/) environment was challenging.

This was solved by converting models into a web API format using Flask, then calling them from [ASP.NET](https://asp.net/).

2-Image Upload and Processing: Ensuring image formats and sizes were handled correctly required testing and preprocessing.

3-Database Synchronization: Keeping patient data, appointments, and predictions linked correctly took careful database design and use of primary/foreign keys.

4-User Interface Usability: Some users found the UI confusing. Based on feedback, we made it simpler by improving labels and error messages**.**

Chapter 5

Testing & Evaluation

* 1. **Testing strategies**

**unit testing:**

Individual functions such as login, registration, and appointment booking were tested to confirm they work correctly

**integration testing**

We tested how the modules work together, such as connecting patient registration with appointment scheduling and billing

**user testing**

Students acted as doctors, patients, and admins to test the system.

Their feedback helped us improve the interface and user experience.

**Machine Learning Model Testing:**

Models were tested using test datasets on Kaggle and Jupyter Notebook to

measure accuracy and consistency before integration.

* 1. **Performance metrics**

-Accuracy of Prediction Models:

Brain Tumor Classification: ~98.02% accuracy on test MRI images.

Skin Cancer Classification: ~88% accuracy on skin image datasets.

Response Time: Image upload and prediction took less than 10 seconds on average.

-Scalability: The system was tested with dummy data for over 200 patients and 20 doctors without slowing down.

-Reliability: The system handled incorrect logins, invalid inputs, and image errors gracefully using validation messages

* 1. **Comparison with existing solutions (if applicable).**

Compared to traditional hospital systems:

Our system adds AI prediction features, which are not common in typical management software.

Many older systems have poor interfaces or are slow.

Our interface is simpler and faster, especially for students and beginner users.

Other AI-based medical tools are often standalone, while our solution combines both hospital management and tumor prediction in one platform.

Chapter 6

Results & Discussion

* 1. **Introduction**

This chapter presents the outcomes from implementing and evaluating the hospital management system. The focus is on the system’s diagnostic performance using machine learning models for skin cancer and brain tumor prediction, as well as how well it fulfilled its intended goals.

Any notable limitations are also discussed.

* 1. **Summary of findings**

The system successfully integrated machine learning models into a user-friendly web interface.

Achieved high accuracy in classification:

Skin Cancer Model: Achieved [insert % if known] accuracy on test images.

Brain Tumor MRI Model: Accurately distinguished between no tumor, glioma, meningioma, and pituitary tumors.

Fast inference times with GPU support ensured smooth clinical usability.

Clinicians were able to upload images via web forms and get instant predictions, demonstrating ease of use and integration.

* 1. **Interpretation of results**

Seamless integration of machine learning in a hospital setting.

Accurate and fast medical image classification.

Real-time prediction via web-based interface.

Helped demonstrate the potential of AI in augmenting diagnostic workflows, especially in resource-limited or high-volume settings.

* 1. **Limitations of the proposed solution**

-Models were trained on publicly available datasets; real-world clinical images may differ.

-No Clinical Validation: The system has not yet been tested or approved in a live clinical environment.

-Lack of Explainability: Models provide predictions without visual or interpretable feedback (e.g., saliency maps).

-Device Dependency: While GPU support improves performance, the system may be slower on CPU-only devices.

Chapter 7

Conclusion & Future Work

* 1. **Summary of contributions**

In conclusion, the Hospital Management System project successfully achieves its main goal of streamlining hospital operations and improving data management. It offers core features such as patient registration, appointment booking, billing, and staff management.

Although there are a few areas for improvement, the system provides a strong foundation for future upgrades and can be scaled into a full-featured hospital automation platform

* 1. **Possible improvements or extensions for future work**

-The graphical user interface is functional but can be made more modern and user-friendly.

- The system lacks real-time notification features such as email or SMS reminders.

-It is currently designed for desktop use only, with no mobile application or responsive web support.

- There is limited automation in certain areas like emergency scheduling or follow-up alerts.

-Security features such as data encryption and multi-factor authentication are not yet implemented.

7.3 **Future Work**

To improve and expand the system, the following enhancements are suggested:

-Develop a mobile application version for easier access by doctors and patients.

-Integrate real-time email or SMS notifications for appointment confirmations and reminders.

-Add advanced reporting features and dashboards for hospital administration.

-Implement data encryption, backups, and user authentication improvements to increase security.

-Enable integration with external systems such as insurance providers or national health databases.

-Support multilingual features to serve a wider range of users.

References

Hospital Management System (APA Style)

Kumar, P., & Agarwal, D. (2015). Hospital management system: A comprehensive review. International Journal of Computer Applications, 119(11), 38–42. https://doi.org/10.5120/21061-4135

Tripathi, A., & Shukla, A. (2014). Implementation of a hospital management system using web technologies. International Journal of Advanced Research in Computer Science and Software Engineering, 4(9), 498–502.

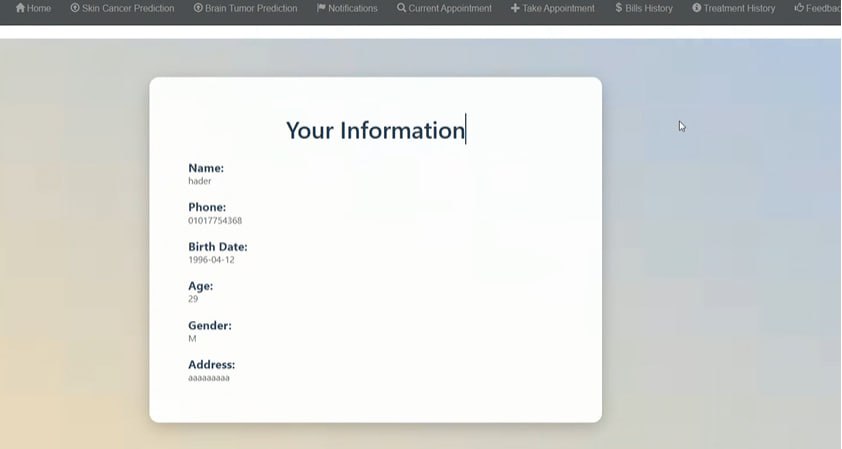
Appendices

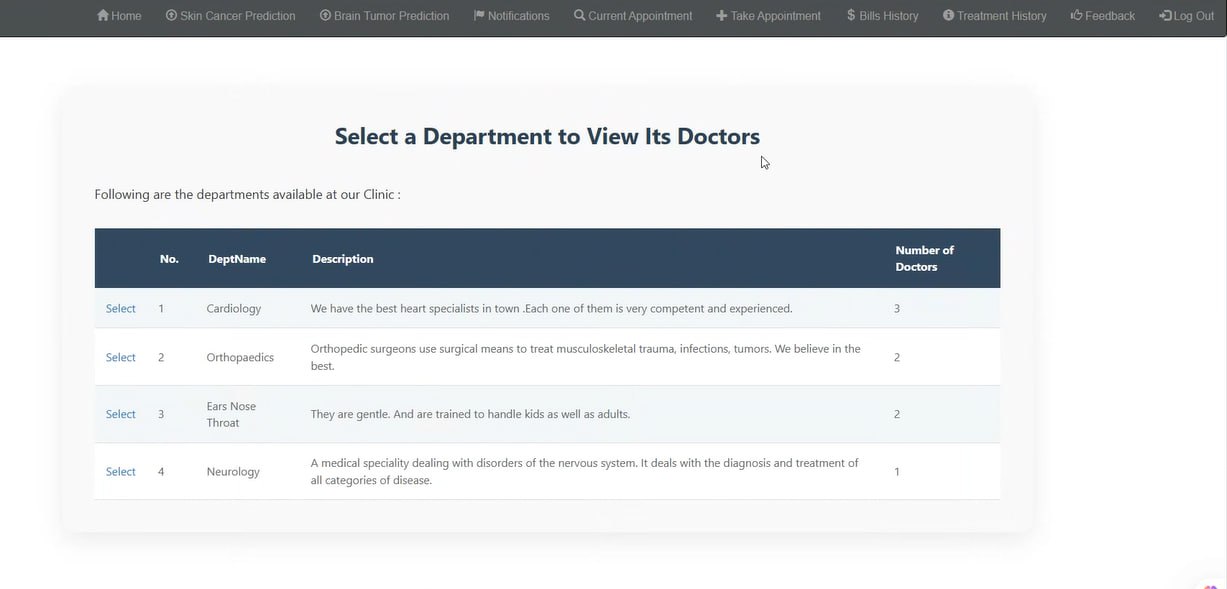
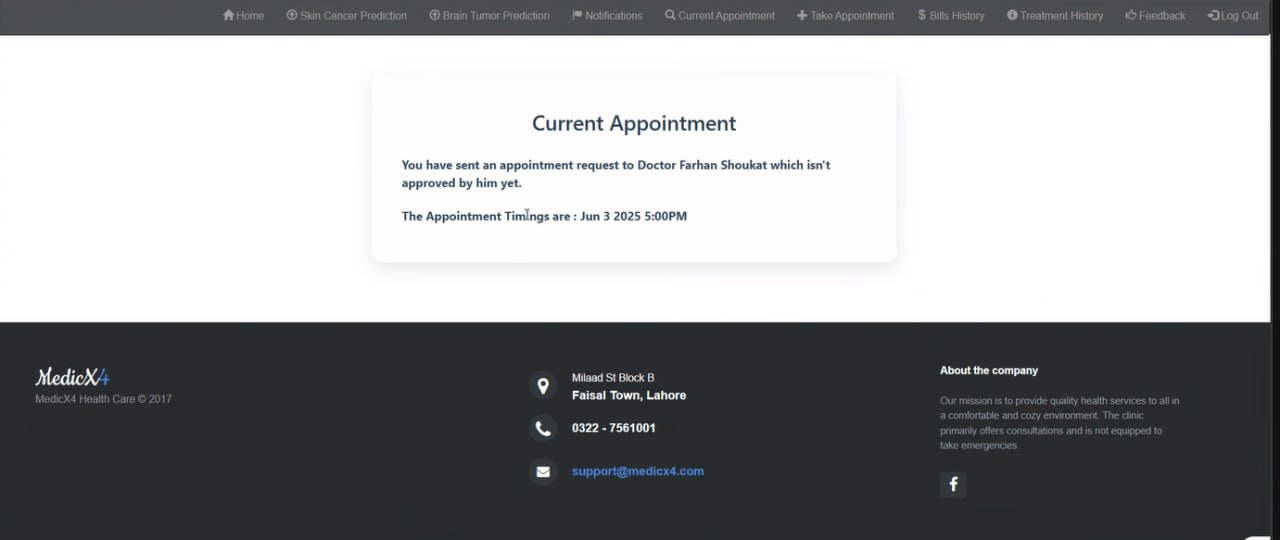
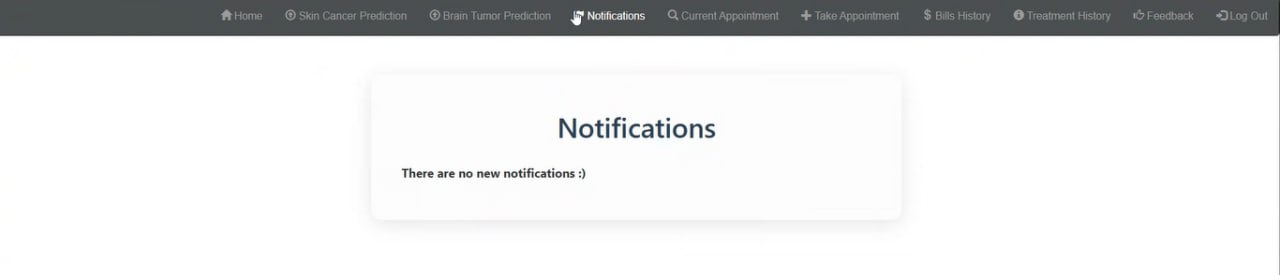
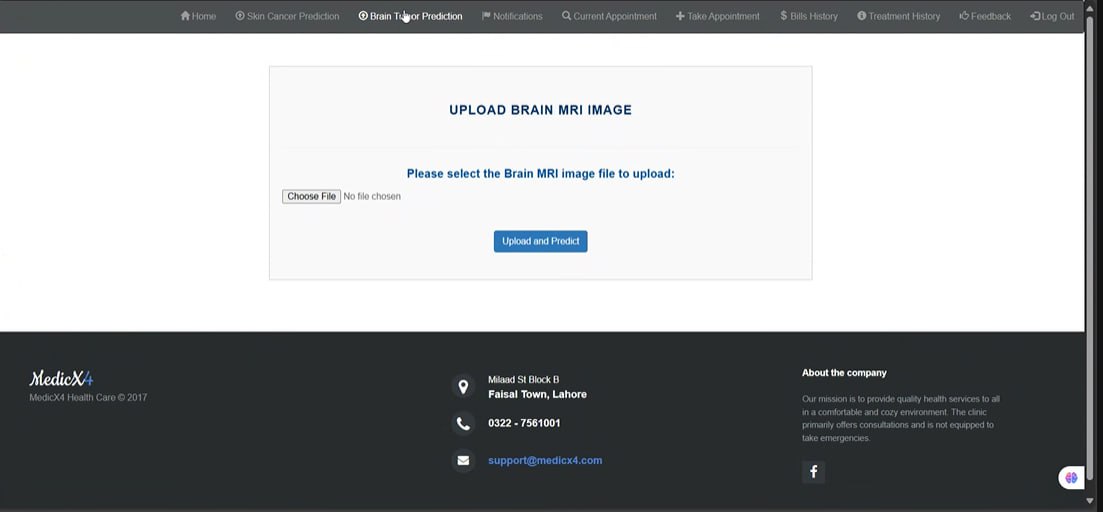
A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a login form

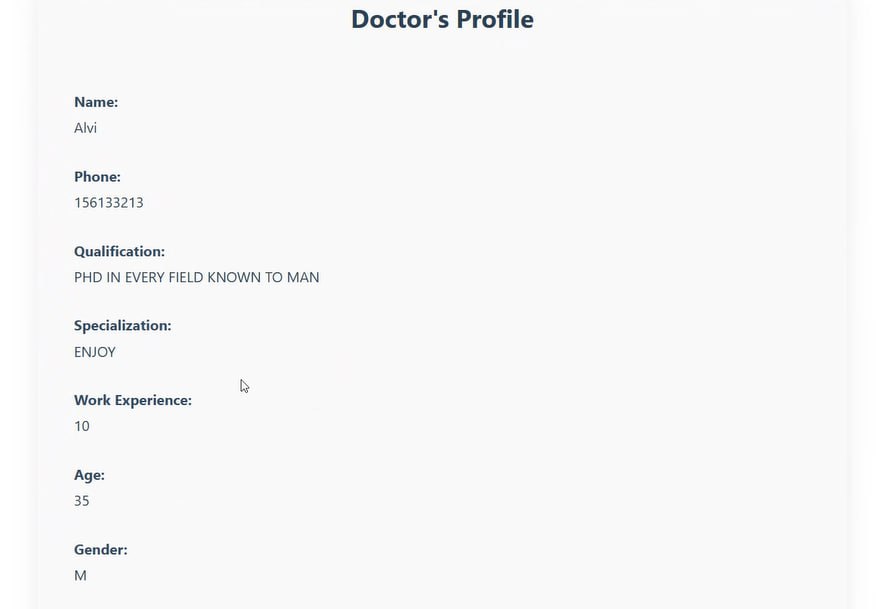
AI-generated content may be incorrect.A screenshot of a login form

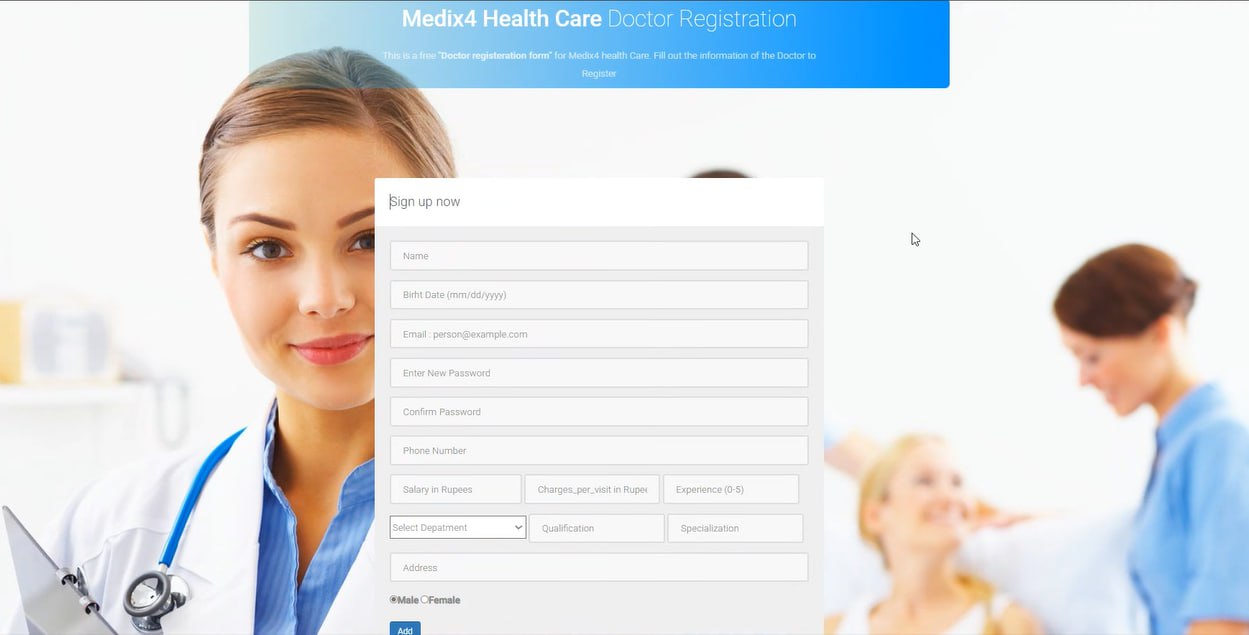
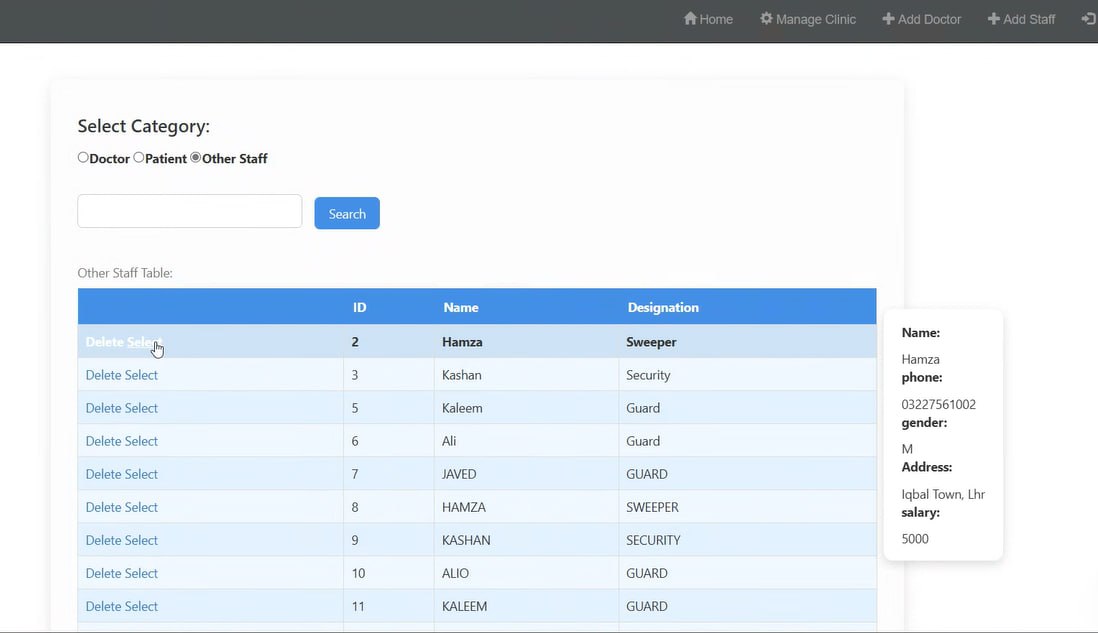
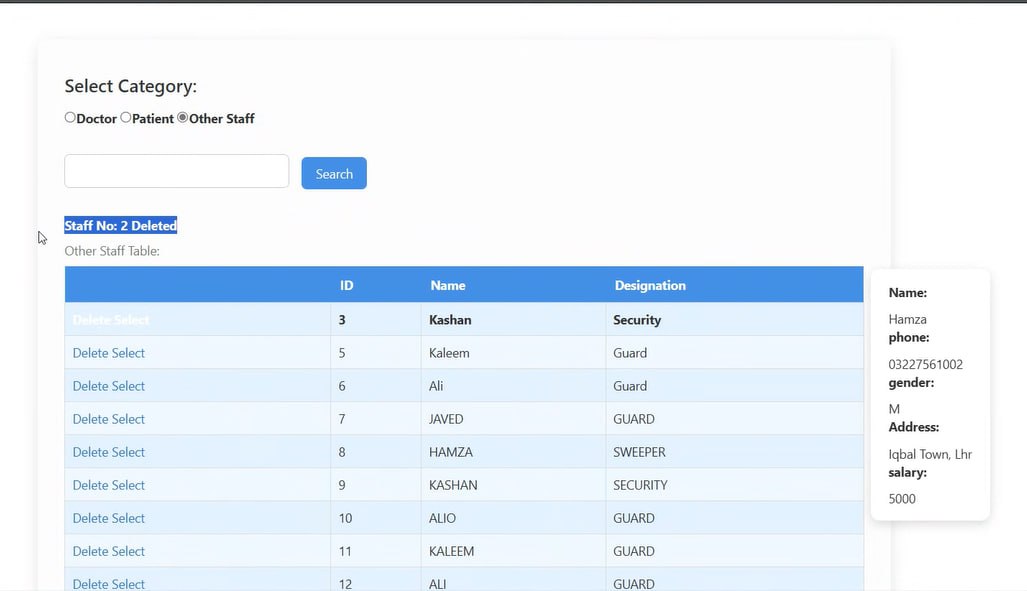
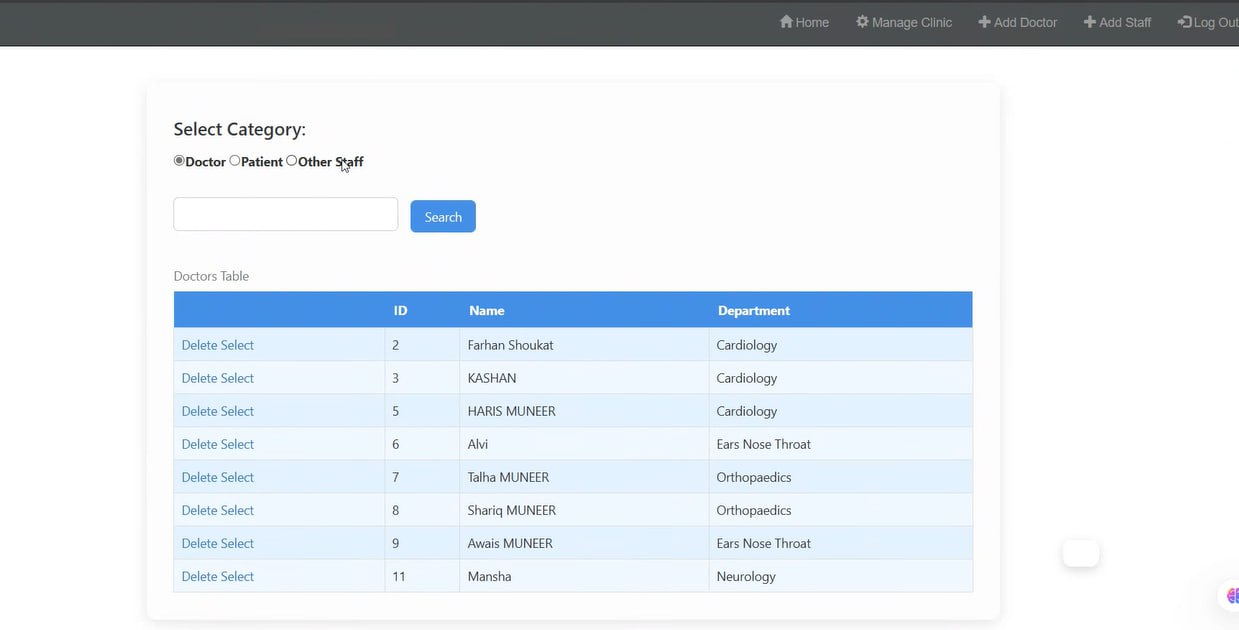
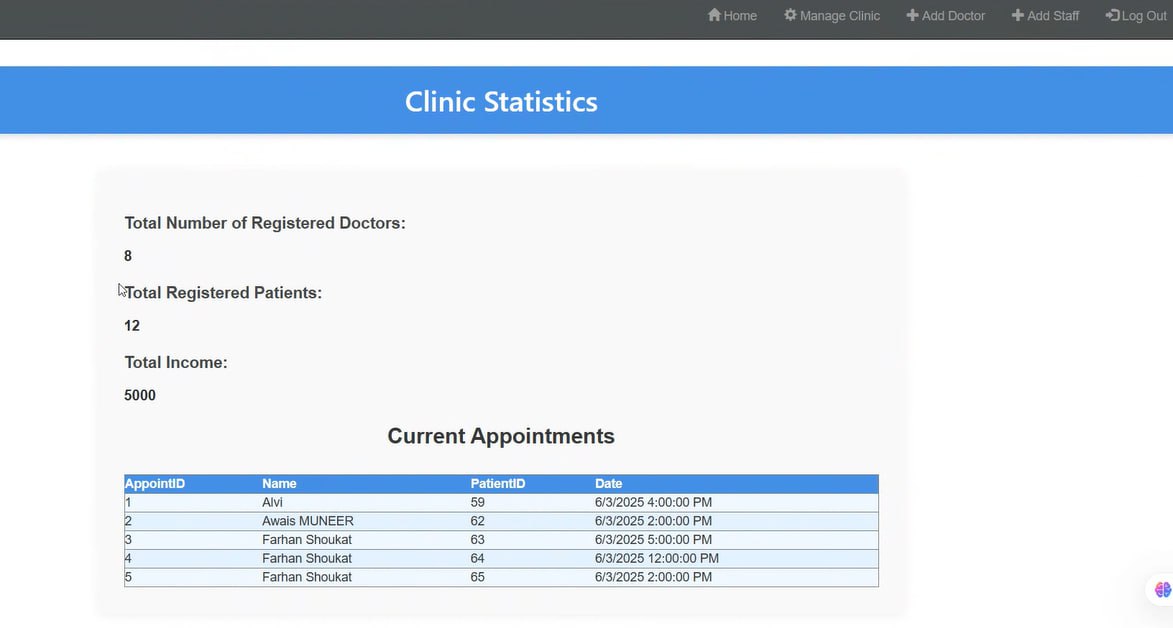
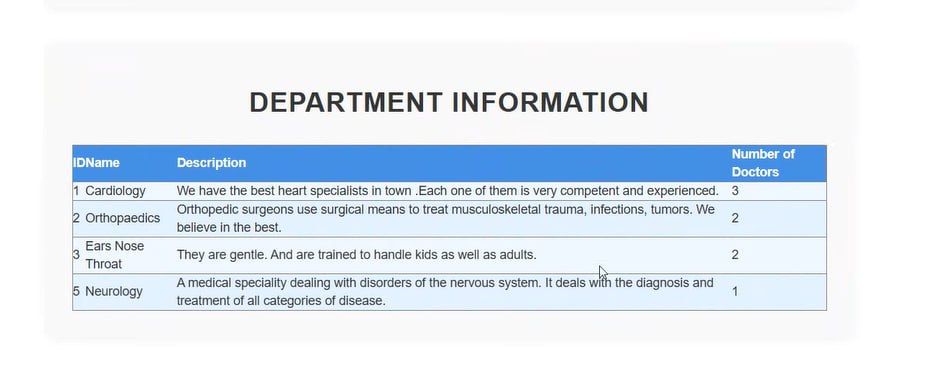
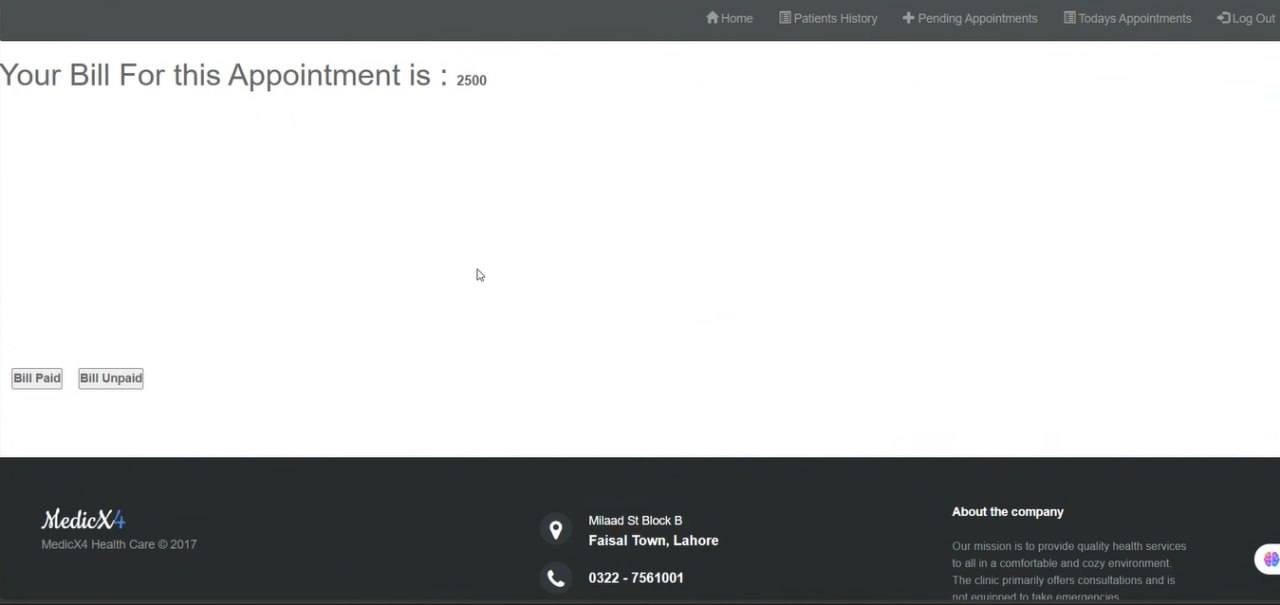
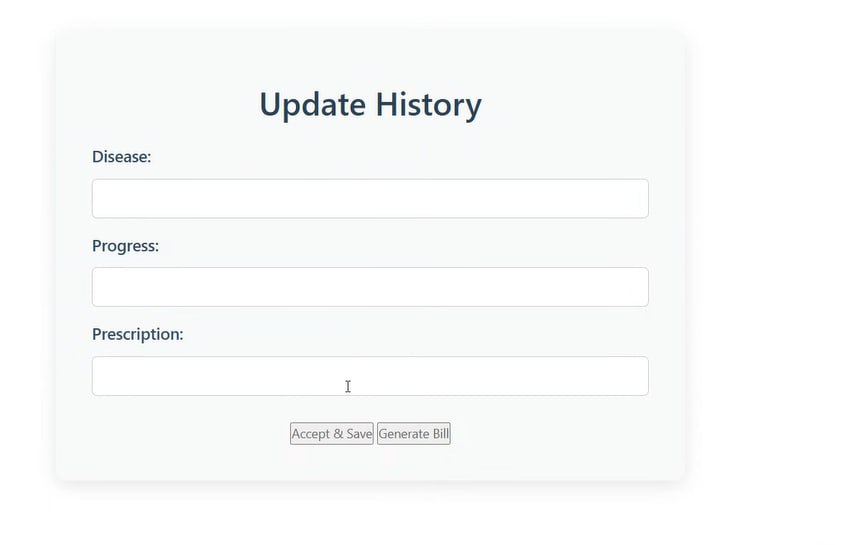
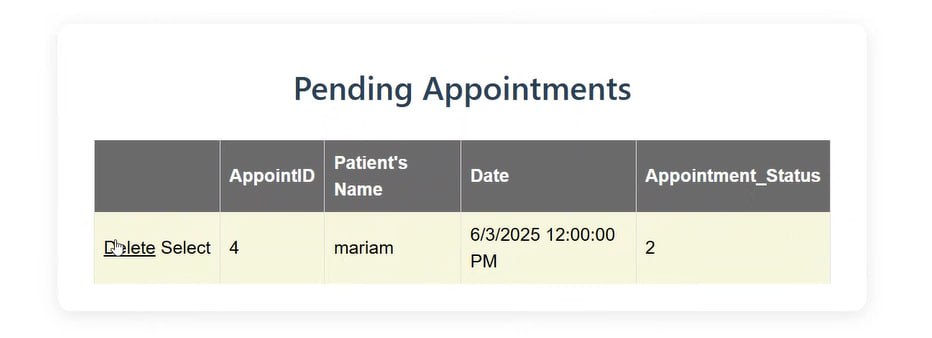
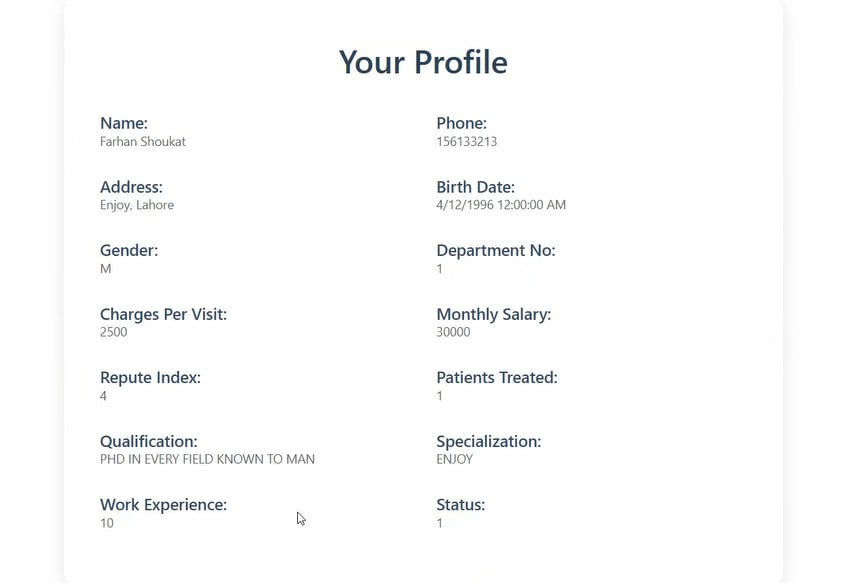
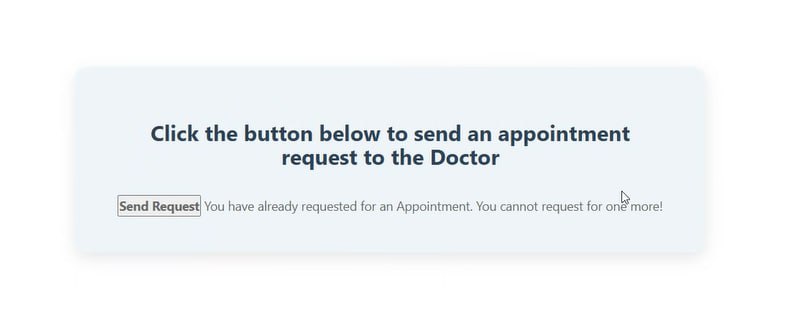
AI-generated content may be incorrect.- Additional diagrams, code snippets, user manuals, or datasets .

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.