**Machine Learning-Based Prediction of PTSD by Analyzing Textual Data**

**Logo, company name

Description automatically generated**

***By***

**Azka 191389**

**Rukhma Riaz 191390**

**Tehreem Jamal 191379**

***Supervisor*Mr. Tariq Naeem**

***Department of Computer Science***

**Air University, Islamabad**

***(2019-2023)***

**Machine Learning-Based Prediction of PTSD by Analyzing Textual Data**

**Logo, company name

Description automatically generated**

***By***

**Azka 191389**

**Rukhma Riaz 191390**

**Tehreem Jamal 191379**

***Supervisor*Mr. Tariq Naeem**

***Bachelor of Science in Information Technology***

**Air University, Islamabad**

***(2019-2023)***

**The candidate confirms that the work submitted is their own and appropriate  
 credit has been given where reference has been made to the work of others**.

**DECLARATION**

We hereby declare that this software, neither whole nor as a part has been copied out from any source. It is further declared that we have developed this software and accompanied report entirely based on our personal efforts. If any part of this project is proved to be copied out from any source or found to be a reproduction of some other, we will stand by the consequences. No Portion of the work presented has been submitted of any application for any other degree or qualification of this or any other university or institute of learning.

Student Name 1 Student Name 2 Student Name 3

Azka Rukhma Riaz Tehreem Jamal

**CERTIFICATE OF APPROVAL**

It is to certify that the final year project of BSIT “Machine Learning-Based Prediction of PTSD by Analyzing Textual Data” was developed by   
**Azka (191389), Rukhma Riaz (191390),** and **Tehreem Jamal (191379)** under the supervision of “Mr. Tariq Naeem” and that in his opinion; it is fully adequate, in scope and quality for the degree of Bachelors of Science in Information Technology.



---------------------------------------

**Supervisor**

---------------------------------------

**Head of Department**

**(Department of Computer Science)**

**Executive Summary**

This document explains the general matters associated with the problem that we're looking to resolve to improve the efficiency of already build tools of PTSD which was 89% but we have achieved it up to 98.14%. The hassle that we're looking to stumble upon is the “prediction of PTSD”. PTSD is a disease that occasionally human beings do now no longer realize that they have. So yeah, it is appropriate for the affected person and society. This document explains the want of this project, heritage and is the reason what are the motives and want which compelled us to finish this project, literature evaluation and is the reason the preceding work associated with this field, ambitions and goal oriented people for this hassle’s solution, like how are we looking to resolve is give an explanation for in procedure, and we've got a described timeline and is the reason all the modules of this project, procedure flow, in addition to how, when, who and why are we going to do what.

We have made web platform of hospital management system in which our PTSD prediction tool is integrated. It is a secure environment as every new doctor and patient requires admin approval to access their dashboard. This system has feature of generating fee bill, downloading fee bill in pdf, and download advice, tests and prescription details. This system verifies credentials before login into platform which is also a plus point for ensuring security. It also creates sessions for more than one user at a time. It provides an online platform to book appointment against doctors.

**Acknowledgement**

All praise is to Almighty Allah who bestowed upon us a minute portion of His boundless knowledge by virtue of which we were able to accomplish this challenging task.

We are greatly indebted to our project supervisor “Mr. Tariq Naeem”. Without their personal supervision, advice and valuable guidance, completion of this project would have been doubtful. We are grateful to them for their encouragement and continual help during this work.

And we are also thankful to our parents and family who have been a constant source of encouragement for us and brought us with the values of honesty & hard work.

Student Name 1 Student Name 2 Student Name 3

Azka Rukhma Riaz Tehreem Jamal

**Abbreviations**

|  |  |
| --- | --- |
| **SRS** | Software Require Specification |
| **PC** | Personal Computer |
| **PTSD** | Post-Traumatic Stress Disorder |
| **MS** | Microsoft |
| **LI** | Limitation |
| **IDE** | Integrated Development Environment |
| **DBMS** | Database Management System |
| **BCS** | Bachelors Computer Science |
| **SQL** | Structured Query Language |
| **UC** | Use Case |
| **UI** | User Interface |
| **FR** | Functional Requirement |
| **CUDA** | Compute Unified Device Architecture |
| **GPU** | Graphics Processing Unit |
| **SSD** | Solid State Drives |
| **HTTP** | Hyper Text Transfer Protocol |
| **FTP** | File Transfer Protocol |
| **SMTP** | Simple Mail Transfer Protocol |
| **MVC** | Model View Controller |
| **ML** | Machine Learning |
| **API** | Application Programming Interface |

**Table of Contents**

[1 Introduction 11](#_Toc87607904)

[1.1 Vision Statement 11](#_Toc87607905)

[1.2 Related System Analysis/Literature Review 11](#_Toc87607906)

[1.3 Project Deliverables 12](#_Toc87607907)

[1.4 System Limitations/Constraints 12](#_Toc87607908)

[1.5 Tools and Technologies 12](#_Toc87607909)

[1.6 Relevance to Course Modules 12](#_Toc87607910)

[2 Problem Definition 13](#_Toc87607911)

[2.1 Problem Statement 13](#_Toc87607914)

[2.2 Problem Solution 13](#_Toc87607915)

[2.3 Objectives of the Proposed System 13](#_Toc87607916)

[2.4 Scope 14](#_Toc87607917)

[2.5 Modules 14](#_Toc87607918)

[2.5.1 Module 1: Module Name 14](#_Toc87607919)

[3 Requirement Analysis 15](#_Toc87607920)

[3.1 User classes and characteristics 15](#_Toc87607922)

[3.2 Requirement Identifying Technique 15](#_Toc87607923)

[3.3 Functional Requirements 15](#_Toc87607924)

[3.3.1 Functional Requirement X 15](#_Toc87607925)

[3.4 Non-Functional Requirements 16](#_Toc87607926)

[3.4.1 Reliability 16](#_Toc87607927)

[3.4.2 Usability 17](#_Toc87607928)

[3.4.3 Performance 17](#_Toc87607929)

[3.4.4 Security 17](#_Toc87607930)

[3.5 External Interface Requirements 17](#_Toc87607931)

[3.5.1 User Interfaces Requirements 17](#_Toc87607932)

[3.5.2 Software interfaces 18](#_Toc87607933)

[3.5.3 Hardware interfaces 18](#_Toc87607934)

[3.5.4 Communications interfaces 18](#_Toc87607935)

[4 Design and Architecture 19](#_Toc87607936)

[4.1 Architectural Design 19](#_Toc87607941)

[4.2 Design Models 19](#_Toc87607942)

[4.3 Data Design 20](#_Toc87607943)

[4.3.1 Data Dictionary 20](#_Toc87607944)

[4.4 Human Interface Design 20](#_Toc87607945)

[4.4.1 Screen Images 20](#_Toc87607946)

[4.4.2 Screen Objects and Actions 20](#_Toc87607947)

[5 Implementation 21](#_Toc87607948)

[5.1 Algorithm 21](#_Toc87607950)

[5.2 External APIs/SDKs 23](#_Toc87607951)

[5.3 User Interface 23](#_Toc87607952)

[5.3.1 Login Screen 5.3.2 Home Screen 24](#_Toc87607953)

[5.3.3 Assignee Dashboard 25](#_Toc87607954)

[5.3.4 New Complaint 25](#_Toc87607955)

[5.4 Deployment 26](#_Toc87607956)

[6 Testing and Evaluation 27](#_Toc87607957)

[6.1 Unit Testing 27](#_Toc87607960)

[6.2 Functional Testing 27](#_Toc87607961)

[6.3 Business Rules Testing 28](#_Toc87607962)

[6.4 Integration Testing 28](#_Toc87607963)

[7 Conclusion and Future Work 29](#_Toc87607964)

[7.1 Conclusion 29](#_Toc87607966)

[7.2 Future Work 29](#_Toc87607967)

[8. References 30](#_Toc87607968)

**List of Figures**

[Figure 1 Use Case Diagram 22](#_Toc135002967)

[Figure 2 Box and Line Diagram 72](#_Toc135002968)

[Figure 3 Architecture Diagram 73](#_Toc135002969)

[Figure 4 Activity Diagram for Admin 74](#_Toc135002970)

[Figure 5 Activity Diagram for Doctor 75](#_Toc135002971)

[Figure 6 Activity Diagram for Patient 76](#_Toc135002972)

[Figure 7 Class Diagram 77](#_Toc135002973)

[Figure 8 Admin Sequence Diagram 78](#_Toc135002974)

[Figure 9 Doctor Sequence Diagram 79](#_Toc135002975)

[Figure 10 Patient Sequence Diagram 80](#_Toc135002976)

[Figure 11 Admin State Transition Diagram 81](#_Toc135002977)

[Figure 12 Doctor State Transition Diagram 82](#_Toc135002978)

[Figure 13 Patient State Transition Diagram 83](#_Toc135002979)

# Introduction

Development of a PTSD predictor is a major turning point in history of physiatrist departments of this world. When a person is not mentally stable, in situations like these; not only that specific human being suffers but also the people around him (Kolk, 2000). Solution to this problem is very vast, that cannot be completed in this project but we are developing PTSD Predictor via Hospital Management System. PTSD basically is a type of depression from which people suffers when they face or experience traumatic events such as war, natural disaster, a serious accident, sexual abuse and any kind of traumatic event that is not easy to forget (Taylor-Desir, 2022). We are using technology to serve humanity by doing the project “Machine Learning-Based Prediction of PTSD by Analyzing Textual Data”. PTSD prediction tool have already been developed but their efficiency level is less than 89%. A platform having less than 89% efficiency cannot be used in hospitals. So, we are developing a web platform which will have efficiency of more than 98.14%. We are using all textual data of PTSD patients, that we got from a researcher and then we have implemented a Machine Learning model on the dataset to train the model so that when we give the data of new patient to it, it can predict whether that patient has PTSD or not. For PTSD checkup, we have developed a proper Hospital Management System which involves Admin, Doctor and Patient. Patient has to book appointment for PTSD checkup and doctor has to run the PTSD test.

## Vision Statement

**For** doctors **who** want to detect PTSD in patients from Hospital Management System online, **the** PTSD Prediction via Hospital Management System **is** an Internet-based web application that **will** help patient to take doctor’s appointment, doctors to fill out questionnaire that will generate PTSD results and manage fee bills. **Unlike** the current PTSD Predictors, hospitals who use **our product** (PTSD Prediction via Hospital Management System) will save doctors time in checking the patients and will help them have a proper system to manage the appointments, fee bills, patients, doctors and keeping the records not manually.

## Related System Analysis/Literature Review

There is no such previous work in Pakistan and any other country in which PTSD predictor is embedded in Hospital Management System via Machine Learning. However, alone PTSD predicators have been made by different techniques like Neural networks, deep learning, voice acoustic measurement and image processing (Mueller-Pfeiffer, Schick, Schulte-Vels, O'Gorman, & Michels, 2013) etc.

Table 1 Related System Analysis with proposed project solution

|  |  |  |
| --- | --- | --- |
| **Application Name** | **Weakness** | **Proposed Project Solution** |
| Predicting Posttraumatic Stress Disorder Risk: A Machine Learning Approach (Wshah, Skalka, & Price, 2019) | Data was collected via smartphone survey. There was no hospital management system built. 7 features were used with limited dataset | Hospital Management System was built to make hospitals buy our web application. Data was collected by primary research by a researcher. We used 10 features. |
| Deep learning- based classification of posttraumatic stress disorder and depression following trauma utilizing visual and auditory markers of arousal and mood (Schultebraucks, Yadav, Shalev, Bonanno, & Galatzer-Levy, 2022) | 89% | 98.1% accuracy |

## Project Deliverables

Following are the deliverables of our project:

* Data collection and preprocessing plan for acquiring PTSD related patient’s data
* Utilizing machine learning methods to identify relevant data features.
* Development of machine learning model for PTSD prediction using the processed textual data.
* Checking accuracy of the model
* Testing and validation of developed PTSD prediction model
* Integration of the developed machine learning model with the Hospital Management System
* Documentation of the project, which involves technical specifications, implementation details, and user manuals etc.
* Evaluation of the project's success
* Conclusion and recommendations for future work.

## System Limitations/Constraints

Following are limitations of the proposed project:

*LI-1: The doctor and patient enter data in the asked format.*

*LI-2: Each user should have his/her level of access to the platform.*

*LI-3: Doctors must not rely 100% on this PTSD platform.*

*LI-4: All textual data used by the application is stored in the form of .csv files.*

*LI-5: All of the records will be saved in the hospital's database by admin even after the patient is fully cured.*

*LI-6: The model trained is not based on large dataset. If the dataset is small or biased, it may affect the model's ability to generalize to new data.*

*LI-7: The Hospital management System contains PTSD predictor embedded in it. So, Hospital who do not deal with Mental disorders cannot use this Hospital Management System.*

## Tools and Technologies

Following are the tools as well as techniques used in implementation of our project:

Table 2 Tools and Technologies for Proposed Project

|  |  |  |  |
| --- | --- | --- | --- |
| **Tools**  **And**  **Technologies** | **Tools** | **Version** | **Rationale** |
| MS Visual Studio | April 2023 (version 1.78) (user setup) | IDE |
| SQLite | Mssql\_django-1.2 | DBMS |
| Jupyter Notebook | CSC 6 | Machine Learning work |
| **Technology** | **Version** | **Rationale** |
| Vs Code | 17.8.1 | Web-end Development |
| Django | 3.0.5 | Back-end Development |
| Starting development server | http://127.0.0.1:8000/ | Local server |
| Python | 3.11.1 | Back-end Development |

## Relevance to Course Modules

Our topic, entitled "Machine Learning-Based Prediction of PTSD by Analysing Textual Data," is connected to several BCS degree-related courses. From Data Structures and Algorithms, Artificial Intelligence and Machine Learning course, we have gained knowledge for training the ML model and choosing best algorithm to implement for better accuracy. For backend and front-end development, we have studied many programming languages such as Python, JavaScript, Django and tools like Visual Studio, SQL Server, Jupyter Notebook etc. and have studied course such as Web Technologies. In our courses on software engineering, database systems, DB administration and management, and web technologies, we have covered software engineering principles such as requirement gathering, system design, testing, as well as deployment, software requirement specification, database management etc. Working with private medical information of patients also presents ethical questions which we have studied in the Professional Practices for IT course.

**2** **Problem Definition**

**2.1 Problem Statement**

Most of the work is done on clinical translation. We did do critical analysis of that clinical translation and after that we come to know that textual data is common among people but the issues faced such as inconsistent data collection, complexity, lack of training, lack of quality assurance etc. are not highlighted. There is lack of mental health awareness. Many people are facing depression, stress, and Post Traumatic Stress Disorder (PTSD). But they are diffferent from each other which can be known by keen observation on the symptoms, which can’t be accurate every time. And patient suffering from PTSD can be treated as stress or depression. Efficiency ratio in existing process was less which has to be improved.

**2.2 Problem Solution**

Based on the problems as discussed earlier in 2.1 we decide to use technology to help doctors using different Machine Learning Models and Frameworks. For this purpose, we made a web-based platform for hospital management system in which our PTSD predictor is integrated whose training was done through ML model. Doctor does one to one session with selected patients with whom he is doubtful about PTSD and ask appropriate questions from patient and fill the questionnaires form by him, to compare it with features that are provided to ML model for training.

* 1. **Objectives of the Proposed System**

BO-1: We have added innovation while checking and detecting PTSD.

BO-2: We have fulfilled the gaps that we can like increasing the accuracy of already built PTSD predictors.

BO-3: We made an online hospital management system embedded with test like PTSD. It links patient, doctor and admin on same platform.

BO-4: We automated prescription which would help to save a record in soft for each patient for future use and also can be downloaded in pdf file.

BO-5: We automated patient discharge process.

BO-6: We automate fee bill generation and downloading facility.

BO-7: We considered this system in security perspective by requiring each new doctor and patient for approval from admin before viewing his dashboard.

BO-8: Our main focus of doing this project was to achieve something related to PTSD prediction efficiency, either it is on research level or development level. According to the latest work only the predictor has been developed but here in this project we have done all of the work from start and then will improve its performance too.

**2.4 Scope**

A web-based system to predict PTSD via Machine Learning Model by using textual data is being defined by this document. A web-based application that is used by doctors for the purpose of prediction of PTSD. PTSD predictor brings great help to doctors in hospitals, like they do not have to take long one-to-one sessions of each and every patient now. It is a customized web application for PTSD which detects symptoms of PTSD in patients by comparing features that are detected during questionnaires session and filled by the doctor in textual form. After successful detection of PTSD, it shows results. It only checks registered patients in which doctor detects something that could be symptom of PTSD. PTSD predictor embedded in a hospital management system equipped the hospital an online, in control website to perform tests like PTSD. Other than this prediction part our website is as a whole a hospital management system in which doctor, patient and admin all have their own dashboard where they can view and have access to those things which are necessary for their role. It is also a secure system in which both new doctor and patient have to get approval from admin before accessing their dashboard. It consists of automated fee bill generation from admin which can be downloaded by patient to pay and also doctor can download the advice, prescription and tests details which he assigns to patient if PTSD is detected.

**2.5 Modules**

Previously developed product was just a PTSD predictor with less than 90% accuracy, no front-end or other things. That is just an experimental study which is not used for medical purposes yet. Now, we are developing a predictor; which can be used in hospitals for the prediction of PTSD. Also, it has a user-friendly web interface which is easy for doctors to use. This platform takes textual data of patients (entered by doctors) and show all the statistics related to the symptoms of PTSD available in the patient. The uniqueness in it is that it is used in hospitals now.

This project has the following six modules.

**2.5.1 Admin Dashboard**

* Admin Signup
* Admin Login
* Recent Doctors
* Recent Patients
* Total Doctors
* Total Patients
* Total Appointments
* Doctor Record
* Register Doctor
* Approve Doctor
* Doctor Specialization
* Patient Record
* Admit Patient
* Approve Patient
* Discharged Patients
* View Appointment
* Book Appointment
* Approve Appointment

**2.5.2 Doctor Dashboard**

* Appointments
* Patients Under Observation
* Discharged Patients
* Delete Appointment
* Your Patient Record
* Your Discharged Patient
* PTSD Test

**2.5.3 Patient Dashboard**

* Doctor details
* View Your Appointment
* Book Appointment
* Download Fee Bill

**2.5.4 PTSD Recommendation system**

* Model Implementation
* Model Training

**2.5.5 PTSD Detection Module**

* Start Test
* Fill Questionnaires
* Get Result
* Advice Patient
* Recommend Tests
* Specify Prescription
* Download Fee Bill
* Submit

**2.5.6 Prescription module**

* Generate Fee Bill
* Download Fee Bill

# Requirement Analysis

Web-based platform for doctors and psychiatrists called the PTSD predictor. Input of symptoms, record search functionality, and patient registration are all necessary. The system needs to operate within predetermined response times and capacity requirements and be dependable and user-friendly. The protection of patient data requires security precautions. A straightforward and consistent user experience should be offered by external interfaces, and the system should work with all required hardware and software tools. Data interchange should be facilitated via efficient communication interfaces.

## User classes and characteristics

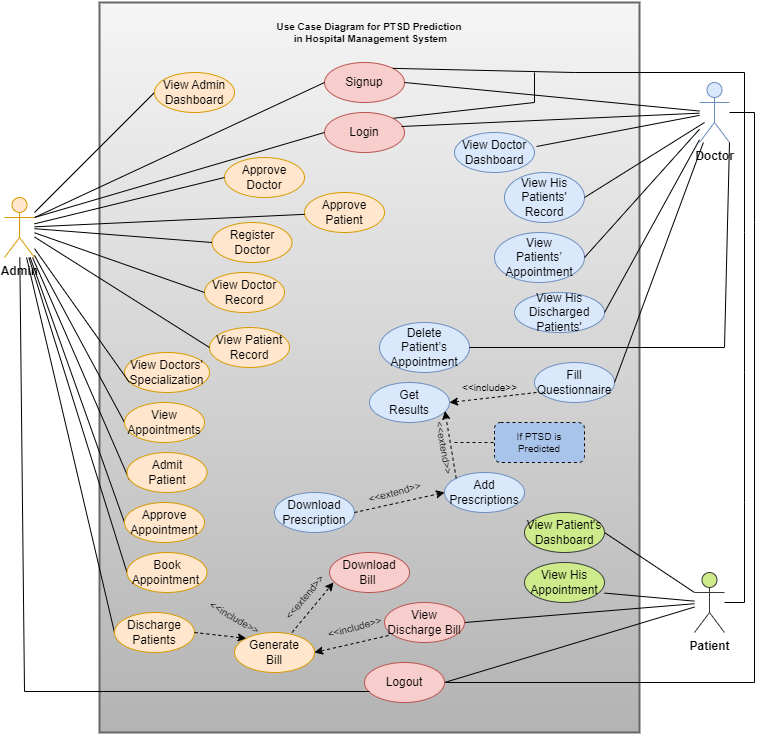
This PTSD predictor is intended to be used by doctors/physiatrists. A typical user should have some experience using web-based platforms. A basic knowledge of browser use is sufficient for the doctors to predict PTSD by this platform.

**Table 3 Show user classes and characteristic for Web-based PTSD Predictor**

|  |  |
| --- | --- |
| **User class** | **Description** |
| **Hospital Admin** | Hospital Admin would have full control to add users (doctors and patient) and provide credentials to them and set access levels accordingly. |
| **Patient** | A patient is someone who has PTSD. He will register on this platform and apply for appointment. |
| **Doctor** | Sometimes there may arrive the situation when someone having mental disorder like PTSD is unaware of it and he goes to doctor for treatment rather than physiatrist, than it can be used by doctor to clear his doubt about what type of treatment the patient needs in actual. To save time and effort, before taking one-to-one sessions the doctor can use this predictor to check are even some symptoms of PTSD in the patient or not. If yes, verified by a predictor then he can take one-to-one sessions of patients for cure purposes. |
| **Developers** | Developers will give the platform to the hospital and update the system simultaneously. |

## Requirement Identifying Technique

Figure 1 Use Case Diagram

****

**Table 4 show UC-1**

|  |  |
| --- | --- |
| **Use Case ID:** | UC-1 |
| **Use Case Name:** | Signup |
| **Actors:** | Admin, doctor, patient |
| **Description:** | Admin sign up by giving his details such as first name, last name, username and password.  Doctor will sign up by giving his details such as first name, last name, username, password, department, contact, address and profile picture.  Patient will sign up by giving his details such as first name, last name, username, password, address, contact, symptoms, doctor name and department he wants to checkup and profile picture. |
| **Trigger:** | The user wants to create his account to get access to the web application. |
| **Preconditions:** | PRE-1: The system is in idle state. |
| **Post conditions:** | POST-1: Admin will login to the application.  POST-2: Doctor and Patient has to wait for the admin’s approval after registering.  POST-3: Admin will give approval.  POST-4: Doctor and Patient can now login to the system. |
| **Normal Flow:** | The admin, doctor and patient will click the signup button on the display screen.  They will then fill the required details.  After that, the login button will appear, Admin will click on it to get access to the system.  Doctor and Patient will wait for the admin’s approval.  When admin will approve them, they can now login to the system. |
| **Alternative Flows:** | None |

**Table 5 show UC-2**

|  |  |
| --- | --- |
| **Use Case ID:** | UC-2 |
| **Use Case Name:** | Login |
| **Actors:** | Admin, doctor, patient |
| **Description:** | Admin, doctor and patient login by giving their details such as, username and password. |
| **Trigger:** | The user wants to create his account to get access to this platform. |
| **Preconditions:** | PRE-1: Users signed up into the system. |
| **Post conditions:** | POST-1: Admin can view his dashboard.  POST-2: admin will approve doctor and patient.  POST-3: Doctor and patient can now login to their respective dashboards. |
| **Normal Flow:** | Admin, doctor and patient will enter username as well as password  After that, click on login button  If credentials found accurate then he/she will have access of the platform  Doctor and patient will enter username and password  A screen will appear that will show them to wait for the approval.  After approval, they can now access their dashboard. |
| **Alternative Flows:** | If the wrong password is entered, users can not enter their respective dashboards. |

**Table 6 show UC-3**

|  |  |
| --- | --- |
| **Use Case ID:** | UC-3 |
| **Use Case Name:** | View Admin Dashboard |
| **Actors:** | Admin |
| **Description:** | Admin can view his dashboard.  On the dashboard, he will view total number of doctors, patients and appointments  There is also box of recent doctors and patients’ information. |
| **Trigger:** | The admin should have his personal dashboard. |
| **Preconditions:** | PRE-1: The admin has logged in to dashboard. |
| **Post conditions:** | POST-1: admin will approve the doctor.  POST-2: admin will approve the patient. |
| **Normal Flow:** | After logging in, admin dashboard will appear. |
| **Alternative Flows:** | None |

**Table 7 show UC-4**

|  |  |
| --- | --- |
| **Use Case ID:** | UC-4 |
| **Use Case Name:** | Approve Doctor |
| **Actors:** | Admin |
| **Description:** | Admin can view credentials of the doctor.  He can delete or approve the doctor. |
| **Trigger:** | No unauthorized doctor can enter the web application. |
| **Preconditions:** | PRE-1: admin has viewed his dashboard. |
| **Post conditions:** | POST-1: admin will approve the patient.  POST-2: admin will admit patient to the hospital. |
| **Normal Flow:** | On the dashboard, there is option of doctor page.  On clicking, he will get 4 options i.e., doctor record, register doctor, approve doctor, doctors’ specialization.  He has to click on “approve doctor” button.  Doctors list will be displayed who are waiting for the approval. |
| **Alternative Flows:** | If the admin deletes the approval, the doctor cannot login. |

**Table 8 show UC-5**

|  |  |
| --- | --- |
| **Use Case ID:** | UC-5 |
| **Use Case Name:** | Approve Patient |
| **Actors:** | Admin |
| **Description:** | Admin can view the credentials of the patient.  He can delete or approve the patient. |
| **Trigger:** | No unauthorized patient can enter the web application. |
| **Preconditions:** | PRE-1: admin has viewed his dashboard.  PRE-2: admin has approved the doctor. |
| **Post conditions:** | POST-1: patient will login  POST-2: patient will book appointment. |
| **Normal Flow:** | On the dashboard, there is option of patient page.  On clicking, he will get 4 options i.e., patient record, admin patient, approve patient and discharge patient.  He has to click on “approve patient” button.  Patients list will be displayed who are waiting for the approval. |
| **Alternative Flows:** | If the admin deletes the approval, the patient cannot login. |

**Table 9 show UC-6**

|  |  |
| --- | --- |
| **Use Case ID:** | UC-6 |
| **Use Case Name:** | Approve Appointment |
| **Actors:** | Admin |
| **Description:** | Admin can view credentials of patient and to which doctor patient want to see.  He can delete or approve the patient’s appointment. |
| **Trigger:** | Admin will not approve the appointment if doctor is not available. |
| **Preconditions:** | PRE-1: admin has viewed his dashboard.  PRE-2: admin has approved the doctor.  PRE-3: Admin has approved the patient. |
| **Post conditions:** | POST-1: patient will login.  POST-2: patient will book appointment. |
| **Normal Flow:** | On the dashboard, there is option of appointment page.  On clicking, he will get 3 options i.e., view appointment, book appointment and approve appointment.  He has to click on “approve appointment” button.  Patients list will be displayed who are waiting for the approval of appointment. |
| **Alternative Flows:** | If the admin deletes the approval, the patient cannot take appointment with the doctor. |

**Table 10 show UC-7**

|  |  |
| --- | --- |
| **Use Case ID:** | UC-7 |
| **Use Case Name:** | Admit Patient |
| **Actors:** | Admin |
| **Description:** | Admin admit the patient to hospital for the checkup. |
| **Trigger:** | This will be done for the patient checkup. |
| **Preconditions:** | PRE-1: admin has approved the patient.  PRE-2: admin has approved the doctor.  PRE-3: admin has approved the appointment. |
| **Post conditions:** | POST-1: appointment will be booked.  POST-2: doctor will check the patient. |
| **Normal Flow:** | On the dashboard, there is option of patient page.  On clicking, he will get 4 options i.e., patient record, admin patient, approve patient and discharge patient.  He has to click on “admit patient” button.  Patients will be admitted |
| **Alternative Flows:** | Nil |

**Table 11 show UC-8**

|  |  |
| --- | --- |
| **Use Case ID:** | UC-8 |
| **Use Case Name:** | Book Appointment |
| **Actors:** | Patient |
| **Description:** | Patient will book appointment to doctor he wants to checkup. |
| **Trigger:** | Without booking appointment, patient cannot get results and prescriptions. |
| **Preconditions:** | PRE-1: admin has approved the patient.  PRE-2: admin has approved the doctor.  PRE-3: admin has approved the appointment.  PRE-4: patient has been admitted. |
| **Post conditions:** | POST-1: appointment will be booked.  POST-2: doctor will start questionnaire.  POST-3: Results will be generated. |
| **Normal Flow:** | Patient will open the web application.  He will click on “book appointment”.  Then, he will add required credentials. |
| **Alternative Flows:** | Admin can also book the appointment but only for patients who have already registered in the system. |

**Table 12 show UC-9**

|  |  |
| --- | --- |
| **Use Case ID:** | UC-9 |
| **Use Case Name:** | View Doctor Dashboard |
| **Actors:** | Doctor |
| **Description:** | Doctor can view his dashboard.  On the dashboard, he will view total number of patients and appointments  There is also box of recent appointments and patients information. |
| **Trigger:** | The doctor should have his personal dashboard. |
| **Preconditions:** | PRE-1: admin has approved the patient.  PRE-2: admin has approved the doctor.  PRE-3: admin has approved the appointment.  PRE-4: patient has been admitted.  PRE-5: appointment has been booked. |
| **Post conditions:** | POST-1: doctor will start questionnaire.  POST-2: Results will be generated. |
| **Normal Flow:** | After logging in, doctor’s dashboard will appear. |
| **Alternative Flows:** | None |

**Table 13 show UC-10**

|  |  |
| --- | --- |
| **Use Case ID:** | UC-10 |
| **Use Case Name:** | View His Patient Record (doctor) |
| **Actors:** | Doctor |
| **Description:** | Doctor will see a table which contains patients list.  List will include patients’ names, profile pictures, symptoms, contact, address, and tests. |
| **Trigger:** | By looking at the list, doctor will the test. |
| **Preconditions:** | PRE-1: admin has approved the patient.  PRE-2: admin has approved the doctor.  PRE-3: admin has approved the appointment.  PRE-4: patient has been admitted.  PRE-5: appointment has been booked. |
| **Post conditions:** | POST-1: doctor will start questionnaire.  POST-2: Results will be generated. |
| **Normal Flow:** | On the dashboard, there is option of patient page.  On clicking, he will get 2 options i.e., your patient record and your discharged patients.  He has to click on “your patient record” button.  A table will be shown of his patients.  He can click on “start” button to start the PTSD test. |
| **Alternative Flows:** | None |

**Table 14 show UC-11**

|  |  |
| --- | --- |
| **Use Case ID:** | UC-11 |
| **Use Case Name:** | Fill Questionnaire |
| **Actors:** | Doctor |
| **Description:** | There will be 11 questions.  In some questions, he will choose options between two.  While in other, doctor will enter the values. |
| **Trigger:** | This is the main purpose of this web application. |
| **Preconditions:** | PRE-1: admin has approved the patient.  PRE-2: admin has approved the doctor.  PRE-3: admin has approved the appointment.  PRE-4: patient has been admitted.  PRE-5: appointment has been booked.  PRE-6: doctor has started the test. |
| **Post conditions:** | POST-1: Results will be generated. |
| **Normal Flow:** | Doctor will fill the 11 questions one by one.  He will then click on “Get Result” button. |
| **Alternative Flows:** | None |

**Table 15 show UC-12**

|  |  |
| --- | --- |
| **Use Case ID:** | UC-12 |
| **Use Case Name:** | Get Results |
| **Actors:** | doctor |
| **Description:** | Result would be generated by the question. It can be either patient has PTSD or not. |
| **Trigger:** | This is the main purpose of this web application. |
| **Preconditions:** | PRE-1: admin has approved the patient.  PRE-2: admin has approved the doctor.  PRE-3: admin has approved the appointment.  PRE-4: patient has been admitted.  PRE-5: appointment has been booked.  PRE-6: doctor has started the test.  PRE-7: doctor has filled the questionnaire. |
| **Post conditions:** | POST-1: if PTSD is detected, doctor will add prescription.  POST-2: Admin will discharge the patient. |
| **Normal Flow:** | On clicking “Get Result” button, results will be generated that whether the patient has PTSD or not. |
| **Alternative Flows:** | None |

**Table 16 show UC-13**

|  |  |
| --- | --- |
| **Use Case ID:** | UC-13 |
| **Use Case Name:** | Add Prescriptions |
| **Actors:** | Doctor |
| **Description:** | Doctor will add prescription which includes advice, test and medicine. He can then download it as a pdf. |
| **Trigger:** | It is mandatory for patients with PTSD. |
| **Preconditions:** | PRE-1: admin has approved the patient.  PRE-2: admin has approved the doctor.  PRE-3: admin has approved the appointment.  PRE-4: patient has been admitted.  PRE-5: appointment has been booked.  PRE-6: doctor has started the test.  PRE-7: doctor has filled the questionnaire.  PRE-8: results have been generated. |
| **Post conditions:** | POST-1: Admin will discharge the patient. |
| **Normal Flow:** | Recommend any test to the patient.  Add any advice/prescription.  Add any medicine.  Click on “submit” button.  Doctor can also download the prescription as pdf. |
| **Alternative Flows:** | If PTSD is not detected in patient, add prescription page will not appear. |

**Table 17 show UC-14**

|  |  |
| --- | --- |
| **Use Case ID:** | UC-14 |
| **Use Case Name:** | Discharge Patient |
| **Actors:** | Admin |
| **Description:** | Admin will discharge patient from hospital. |
| **Trigger:** | Admin wants to generate the bill. |
| **Preconditions:** | PRE-1: admin has approved the patient.  PRE-2: admin has approved the doctor.  PRE-3: admin has approved the appointment.  PRE-4: patient has been admitted.  PRE-5: appointment has been booked.  PRE-6: doctor has started the test.  PRE-7: doctor has filled the questionnaire.  PRE-8: results has been generated.  PRE-9: prescription have been added. |
| **Post conditions:** | POST-1: patient can view the discharge bill. |
| **Normal Flow:** | On the dashboard, there is option of patient page.  On clicking, he will get 4 options i.e., patient record, admin patient, approve patient and discharge patient.  He has to click on “discharge patient” button to discharge the patient. |
| **Alternative Flows:** | None |

**Table 18 show UC-15**

|  |  |
| --- | --- |
| **Use Case ID:** | UC-15 |
| **Use Case Name:** | Generate Bill |
| **Actors:** | Admin |
| **Description:** | Admin will generate bill for patient. |
| **Trigger:** | Patient have to pay for the admission in the hospital. |
| **Preconditions:** | PRE-1: admin has approved the patient.  PRE-2: admin has approved the doctor.  PRE-3: admin has approved the appointment.  PRE-4: patient has been admitted.  PRE-5: appointment has been booked.  PRE-6: doctor has started the test.  PRE-7: doctor has filled the questionnaire.  PRE-8: results have been generated.  PRE-9: prescription have been added.  PRE-10: patient have been discharged. |
| **Post conditions:** | POST-1: patient can view the discharge bill. |
| **Normal Flow:** | Admin will click on “discharge patient” button to discharge the patient.  A bill will be generated.  Admin will add room charged, doctor fee, medicine cost and other charges.  Admin will view the bill with total cost.  Admin will save the bill and can download as pdf. |
| **Alternative Flows:** | None |

**Table 19 show UC-16**

|  |  |
| --- | --- |
| **Use Case ID:** | UC-16 |
| **Use Case Name:** | View Patients Dashboard |
| **Actors:** | Patient |
| **Description:** | Patient can view his dashboard.  On the dashboard, he will view doctor’s name, symptoms, doctor contact, doctor address, doctor department, and admit date. |
| **Trigger:** | The patient should have his personal dashboard. |
| **Preconditions:** | PRE-1: admin has approved the patient.  PRE-2: admin has approved the doctor.  PRE-3: admin has approved the appointment.  PRE-4: patient has been admitted.  PRE-5: appointment has been booked.  PRE-6: doctor has started the test.  PRE-7: doctor has filled the questionnaire.  PRE-8: results have been generated.  PRE-9: prescription have been added.  PRE-10: patient have been discharged.  PRE-11: discharge bill has been generated. |
| **Post conditions:** | POST-1: patient will view his discharged bill  POST-2: logout |
| **Normal Flow:** | After logging in, patient dashboard will appear with doctor’s name, symptoms, doctor contact, doctor address, doctor department, and admit date. |
| **Alternative Flows:** | None |

**Table 20 show UC-17**

|  |  |
| --- | --- |
| **Use Case ID:** | UC-17 |
| **Use Case Name:** | View Discharged Bill |
| **Actors:** | Patient |
| **Description:** | Patient will view discharged bill created by the admin. |
| **Trigger:** | Patient has to pay the bill. |
| **Preconditions:** | PRE-1: admin has approved the patient.  PRE-2: admin has approved the doctor.  PRE-3: admin has approved the appointment.  PRE-4: patient has been admitted.  PRE-5: appointment has been booked.  PRE-6: doctor has started the test.  PRE-7: doctor has filled the questionnaire.  PRE-8: results have been generated.  PRE-9: prescription have been added.  PRE-10: patient have been discharged.  PRE-11: discharge bill has been generated. |
| **Post conditions:** | POST-1: logout |
| **Normal Flow:** | Patient will click on discharge button on the left side of the patient dashboard.  Patient will view the bill picture with all the details and total amount to be paid. |
| **Alternative Flows:** | None |

**Table 21 show UC-18**

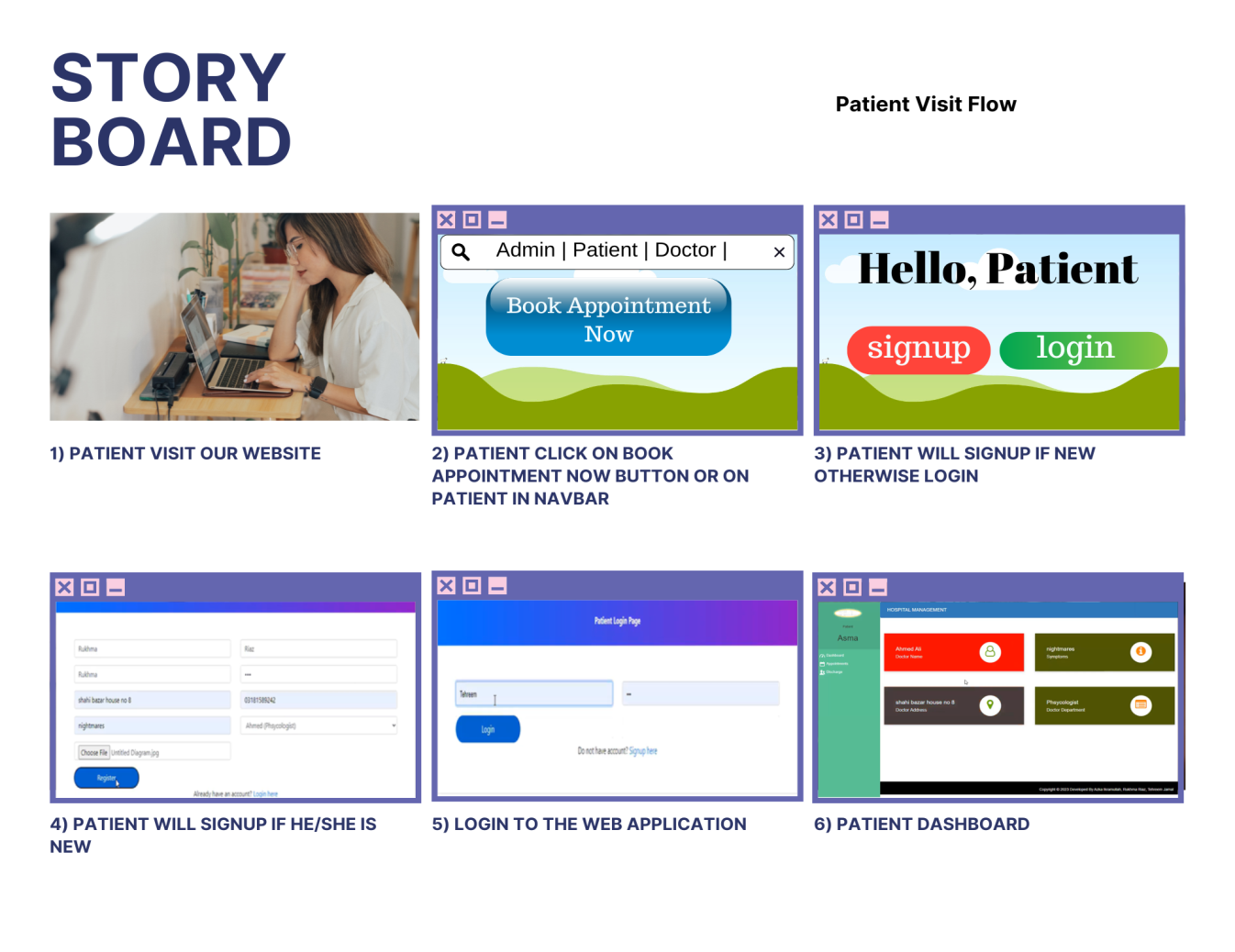
|  |  |
| --- | --- |
| **Use Case ID:** | UC-18 |
| **Use Case Name:** | Logout |
| **Actors:** | Admin, doctor, patient |
| **Description:** | Admin, doctor and patient will click on logout button. |
| **Trigger:** | It is must if users are not using the application. |
| **Preconditions:** | PRE-1: checkup has been done. |
| **Post conditions:** | Nil |
| **Normal Flow:** | Admin, doctor and patient will click on “Logout” button on the right top corner of the bar.  It will redirect them to the main web application page. |
| **Alternative Flows:** | Keep on using the web application. |

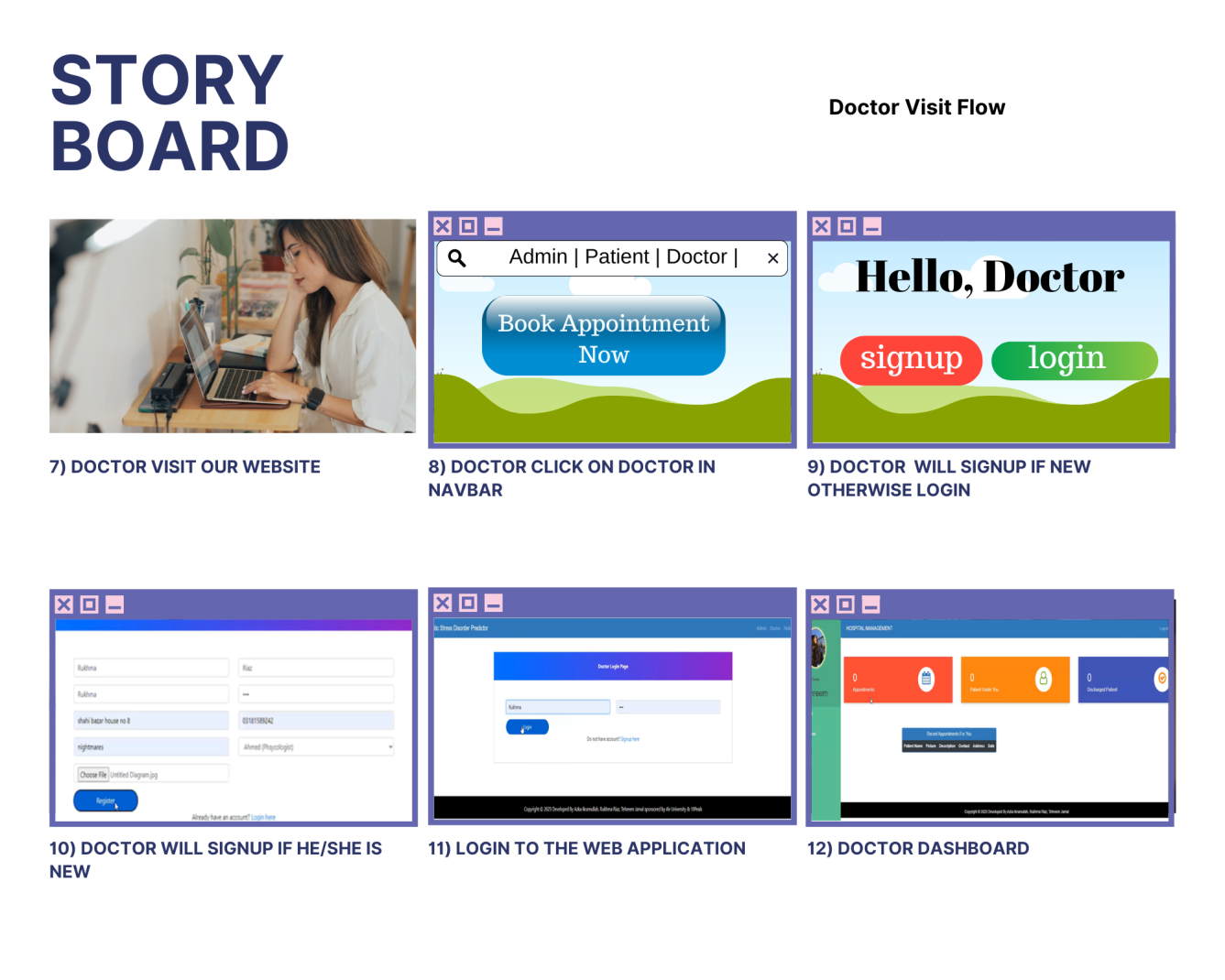
**Event- response table is for real-time system in which most of the functionalities are performed at backend.**

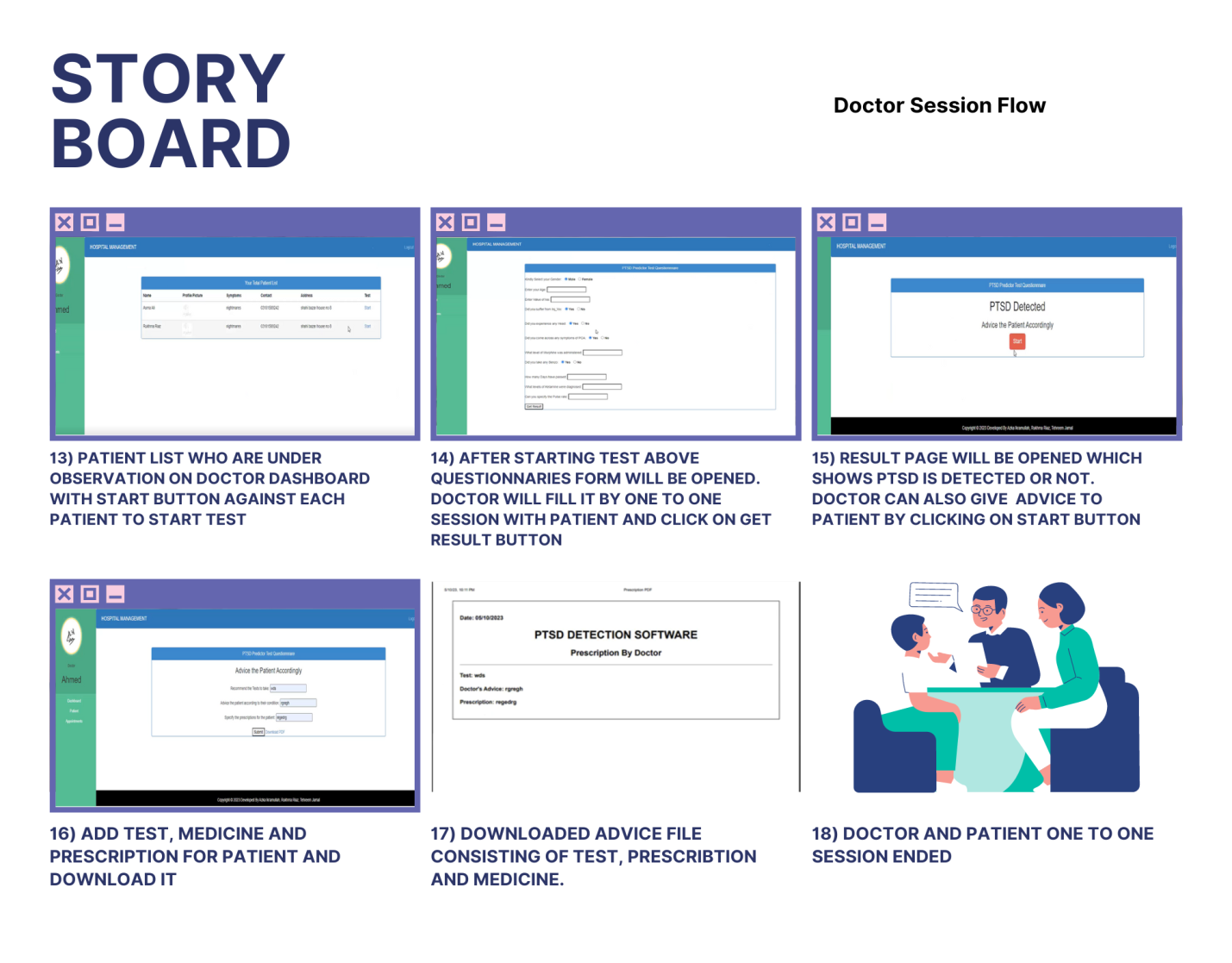
**Table 22 Event- Response Table**

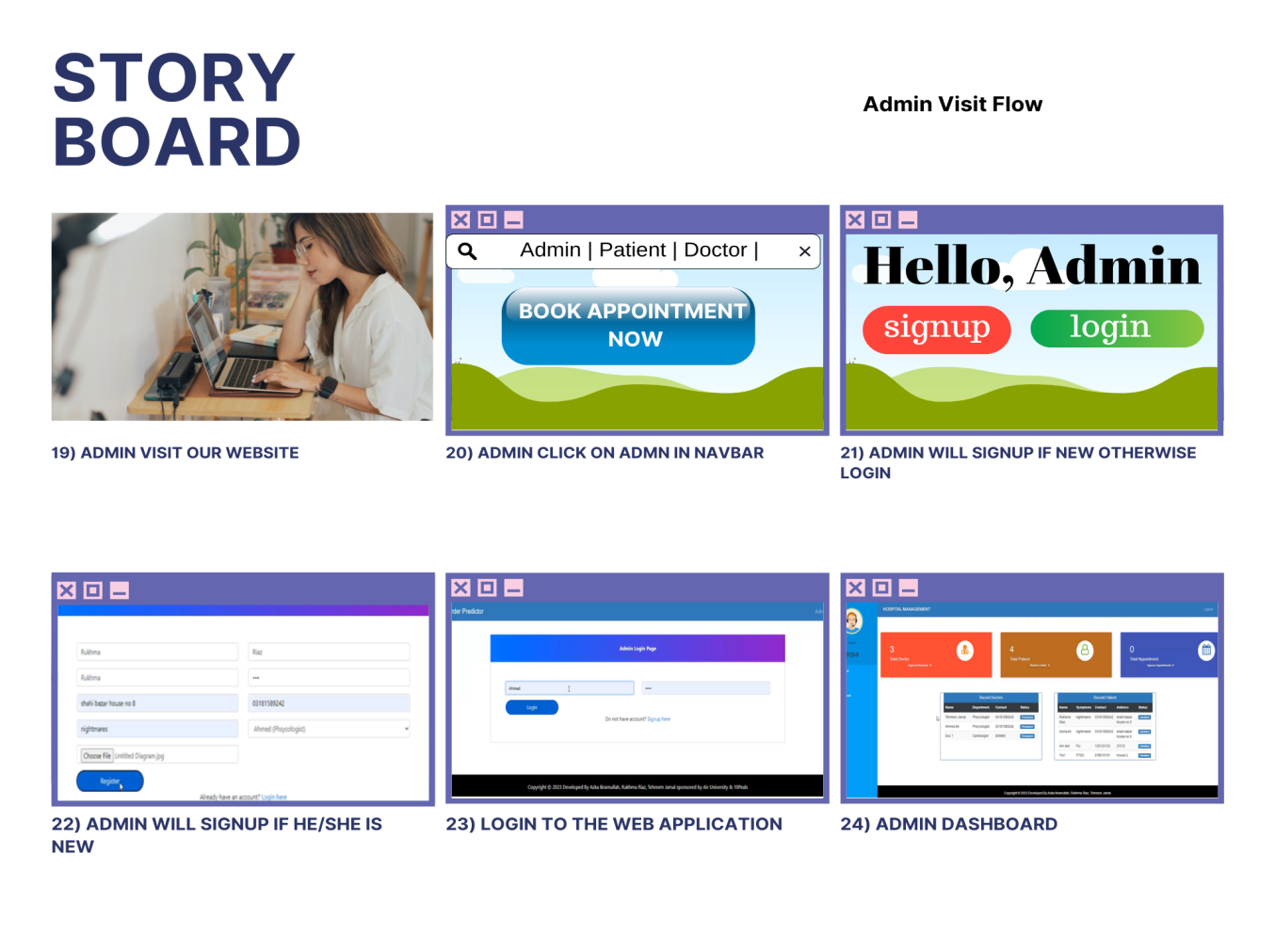
|  |  |  |
| --- | --- | --- |
| **Event** | **System State** | **Response** |
| New users have to sign up by giving required details according to his role. | All provided details will be verified from system whether user already exists or not. If new then details will be stored into system. | User can now login into the system by already set credentials. |
| Login and password will be required. | If wrong password is entered, login is not preceded. | Wait few seconds until the page is loading and new page is open. |
| Admin signup if he is new otherwise login. | Admin will be verified that he entered correct credentials or not. | If admin ender correct credentials then admin dashboard will be opened otherwise, he will not be able to access dashboard. |
| Doctor apply into hospital | If doctor already exit no request will be forwarded. | Doctor Request will be send to admin for approval. |
| Admin approve doctor | Approval send to doctor | Doctor can access his/her dashboard |
| Patient register into system | If patient already exist no request will be forwarded | Patient Request will be send to admin for approval. |
| Admin approve patient | Approval send to patient | Patient can access his/her dashboard |
| Admin assign doctor to patient. | One patient can be assigned to one doctor only, so it will be verified weather that patient is already assigned to other doctor or not. | Doctor can have session with patient. |
| Doctor click on Start button to open questionnaires form | Questionnaires form will be loaded | Questionnaires form will be opened |
| Doctor start asking questions with patient and filling the form and click on get result button. | PTSD prediction would be processed in backend. | Result that weather he/she have PTSD or not will be displayed. |
| Doctor adds prescription, medication and test if PTSD is detected and click on download button. | Downloading will be processed. | A downloaded pdf file of prescription, medication and tests will be saved in system. |
| Doctor discharge patient. | Patient name will be send to discharged patient list and removed from patient under you list. | Admin Generate fee bill. |
| Admin generate fee bill. | Fee bill send to patient dashboard. | Patient download fee bill and pay it. |
| PTSD checker will process the doctor’s personal details added in the web page. | Entering wrong format will lead to the error and there will be no further proceeding. | Loading the page to open next page. |

**Storyboarding for graphically intensive applications.**

****

****

****

****

## Functional Requirements

### Functional Requirement X

Table 23 Description of FR-1

|  |  |
| --- | --- |
| **Identifier** | FR-1 |
| **Title** | Admin Signup |
| **Requirement** | If written in a **user perspective**  The Admin shall be able to enter first name, last name and password to enter the Admin dashboard.  If written in a **system perspective**  After clicking submit button, the system shall save admin’s username and password. |
| **Source** | This requirement comes from hospital admin. |
| **Rationale** | To make sure any unknown person other than admin does not enter.  To access the Hospital Management System and supervise it. |
| **Dependencies** | All other FR are dependent on FR-1. |
| **Priority** | High |

Table 24 Description of FR-2

|  |  |
| --- | --- |
| **Identifier** | FR-2 |
| **Title** | Doctor Registration |
| **Requirement** | If written in a **user perspective**  The doctor shall be able to enter first name, last name, user, password, department, contact, address and picture to register himself and enter doctor’s dashboard.  If written in a **system perspective**  After clicking register button, the system shall save doctor’s credentials. |
| **Source** | This requirement comes from hospital doctor. |
| **Rationale** | To make sure any unknown person other than doctor does not enter.  To access the Hospital Management System and check PTSD patients. |
| **Dependencies** | All other FR are dependent on FR-2 |
| **Priority** | High |

Table 25 Description of FR-3

|  |  |
| --- | --- |
| **Identifier** | FR-3 |
| **Title** | Patient Registration |
| **Requirement** | If written in a **user perspective**  The Patient shall be able to enter first name, last name, user, password, address, contact number, symptoms, choose doctor’s name and department and add a picture to register himself and enter patient dashboard.  If written in a **system perspective**  After clicking register button, the system shall save patient’s credentials. |
| **Source** | This requirement comes from patient. |
| **Rationale** | To make sure any unknown person other than patient does not enter.  For patient’s checkup. |
| **Dependencies** | All other FR are dependent on FR-3 |
| **Priority** | High |

Table 26 Description of FR-4

|  |  |
| --- | --- |
| **Identifier** | FR-4 |
| **Title** | Admin Login |
| **Requirement** | If written in a **user perspective**  The admin shall be able to enter user and password to enter the admin’s dashboard.  If written in a **system perspective**  After clicking login button, system shall take admin to the admin dashboard. |
| **Source** | This requirement comes from hospital admin. |
| **Rationale** | To have access to the admin dashboard. |
| **Dependencies** | FR-1 |
| **Priority** | High |

Table 27 Description of FR-5

|  |  |
| --- | --- |
| **Identifier** | FR-5 |
| **Title** | Doctor Login |
| **Requirement** | If written in a **user perspective**  The doctor shall be able to enter user and password to enter the doctor’s dashboard.  If written in a **system perspective**  After clicking login button, system shall take doctor to the approval page. |
| **Source** | This requirement comes from hospital doctor. |
| **Rationale** | To have access to doctor dashboard and perform the functionalities. |
| **Business Rule (if required)** | The doctor cannot enter doctor’s dashboard until the admin approves the registered doctor. |
| **Dependencies** | FR-2 |
| **Priority** | High |

Table 28 Description of FR-6

|  |  |
| --- | --- |
| **Identifier** | FR-6 |
| **Title** | Patient login |
| **Requirement** | If written in a **user perspective**  The patient shall be able to enter user and password to enter the patient dashboard.  If written in a **system perspective**  After clicking login button, system shall take patient to the approval page. |
| **Source** | This requirement comes from patient. |
| **Rationale** | To access the patient dashboard and access patient functionalities. |
| **Business Rule (if required)** | The patient cannot enter patient’s dashboard until the admin approves the registered patient. |
| **Dependencies** | FR-3 |
| **Priority** | High |

Table 29 Description of FR-7

|  |  |
| --- | --- |
| **Identifier** | FR-7 |
| **Title** | View doctor’s record |
| **Requirement** | If written in a **user perspective**  The admin shall be able to view doctors’ records that are saved in the system.  If written in a **system perspective**  After clicking on doctor record box, the system will view the saved doctor credentials. |
| **Source** | This requirement comes from hospital admin. |
| **Rationale** | If admin want to search for a doctor, you can view doctors record. |
| **Dependencies** | FR-2 |
| **Priority** | Low |

Table 30 Description of FR-8

|  |  |
| --- | --- |
| **Identifier** | FR-8 |
| **Title** | Register doctor |
| **Requirement** | If written in a **user perspective**  The admin shall be able to register the doctor by himself too.  If written in a **system perspective**  After registering, the system shall save the doctors credentials and doctor can login then. |
| **Source** | This requirement comes from hospital admin. |
| **Rationale** | Admin can register any doctor without the need of giving approval. |
| **Dependencies** | Nil |
| **Priority** | Medium |

Table 31 Description of FR-9

|  |  |
| --- | --- |
| **Identifier** | FR-9 |
| **Title** | Approve doctor |
| **Requirement** | If written in a **user perspective**  The admin shall be able to approve doctor, so that the doctor can successfully login to the dashboard.  If written in a **system perspective**  After clicking approve box, the system shall automatically save doctors credentials. |
| **Source** | This requirement comes from hospital admin. |
| **Rationale** | No suspicious person can enter as a doctor. |
| **Business Rule (if required)** | On clicking reject box, the doctor cannot login to the doctor’s dashboard. |
| **Dependencies** | FR-5 |
| **Priority** | High |

Table 32 Description of FR-10

|  |  |
| --- | --- |
| **Identifier** | FR-10 |
| **Title** | View doctor specialization |
| **Requirement** | If written in a **user perspective**  The admin shall be able to view doctor’s specialization. |
| **Source** | This requirement comes from hospital admin. |
| **Rationale** | Admin can edit it when needed. |
| **Dependencies** | Nil |
| **Priority** | Low |

Table 33 Description of FR-11

|  |  |
| --- | --- |
| **Identifier** | FR-11 |
| **Title** | View patients record |
| **Requirement** | If written in a **user perspective**  The admin shall be able to view all patients records that are under every doctor.  If written in a **system perspective**  On clicking patient record box, the system shall display patient’s name, picture, address, contact, symptoms. |
| **Source** | This requirement comes from hospital admin. |
| **Rationale** | If admin want to search for a patient, you can view patients record. |
| **Business Rule (if required)** | Admin can also update the patients record and delete the patient too. |
| **Dependencies** | Nil |
| **Priority** | Low |

Table 34 Description of FR-12

|  |  |
| --- | --- |
| **Identifier** | FR-12 |
| **Title** | Admit patient |
| **Requirement** | If written in a **user perspective**  The admin shall be able to admit the patient to the hospital.  If written in a **system perspective**  After clicking on admit patient, the system shall display \_? |
| **Source** | This requirement comes from hospital admin. |
| **Rationale** | Admin can admit patient to make his appointment to the doctor. |
| **Dependencies** | FR-3 |
| **Priority** | High |

Table 35 Description of FR-13

|  |  |
| --- | --- |
| **Identifier** | FR-13 |
| **Title** | Approve patients |
| **Requirement** | If written in a **user perspective**  The admin shall be able to approve the registered patient.  If written in a **system perspective**  After clicking on approve button, the system shall save the registered patient’s user and credentials. |
| **Source** | This requirement comes from hospital admin. |
| **Rationale** | No suspicious person can enter as a patient. |
| **Dependencies** | FR-3 |
| **Priority** | High |

Table 36 Description of FR-14

|  |  |
| --- | --- |
| **Identifier** | FR-14 |
| **Title** | Discharge patients |
| **Requirement** | If written in a **user perspective**  The admin shall be able to discharge the patient to empty the hospital bed.  If written in a **system perspective**  On clicking discharge button, the system shall pop up page to enter the discharge bill. |
| **Source** | This requirement comes from hospital admin. |
| **Rationale** | To clear the bills of the patient. |
| **Dependencies** | FR-15, FR-16, FR-40 |
| **Priority** | High |

Table 37 Description of FR-15

|  |  |
| --- | --- |
| **Identifier** | FR-15 |
| **Title** | Generate bill |
| **Requirement** | If written in a **user perspective**  The admin shall be able to generate bill to handover to the patient.  If written in a **system perspective**  After writing amounts, the system shall add the amount and give total bill. |
| **Source** | This requirement comes from hospital admin. |
| **Rationale** | To show the bills to patients. |
| **Dependencies** | FR-16, FR-40 |
| **Priority** | Medium |

Table 38 Description of FR-16

|  |  |
| --- | --- |
| **Identifier** | FR-16 |
| **Title** | Download bill |
| **Requirement** | If written in a **user perspective**  The admin shall be able to download the bill as a pdf.  The patient can view and download the bill as a pdf too.  If written in a **system perspective**  After clicking the download option, the system shall download the bill to the PC.  the system will show the generated bill to the patient’s dashboard after the admin will generate the bill. |
| **Source** | This requirement comes from hospital admin and patient. |
| **Rationale** | To save the bill on your PC for the record. |
| **Dependencies** | Nil |
| **Priority** | Low |

Table 39 Description of FR-17

|  |  |
| --- | --- |
| **Identifier** | FR-17 |
| **Title** | View appointments |
| **Requirement** | If written in a **user perspective**  The admin shall be able to view appointments of every patient.  If written in a **system perspective**  On clicking view appointment, the system shall view the doctors name, patient’s info and date of appointment. |
| **Source** | This requirement comes from hospital admin. |
| **Rationale** | To view appointments when needed. |
| **Dependencies** | Nil |
| **Priority** | Low |

Table 40 Description of FR-18

|  |  |
| --- | --- |
| **Identifier** | FR-18 |
| **Title** | Book appointment |
| **Requirement** | If written in a **user perspective**  The admin shall be able to book the patient’s appointment with the required department’s doctor.  The patient shall be able to book the appointment when open the web application.  If written in a **system perspective**  On clicking book appointment, the system shall view \_? |
| **Source** | This requirement comes from hospital admin, patient. |
| **Rationale** | So that patient can get checkup from the doctor. |
| **Dependencies** | FR-17, FR-31, FR-32, FR-33, FR-34, FR-35, FR-37, FR-38, and FR-39 |
| **Priority** | High |

Table 41 Description of FR-19

|  |  |
| --- | --- |
| **Identifier** | FR-19 |
| **Title** | Approve appointment |
| **Requirement** | If written in a **user perspective**  The admin shall be able to approve the registered patient’s appointment.  If written in a **system perspective**  After clicking on approve button, the system shall save the appointment and display it to the doctor. |
| **Source** | This requirement comes from hospital admin. |
| **Rationale** | If the doctor with required department is available, the admin will approve the appointment. |
| **Business Rule (if required)** | If admin rejects the appointment, patient have to book appointment again. |
| **Dependencies** | FR-31, FR-32, FR-33, FR-34, FR-35, FR-37, FR-38 and FR-39 |
| **Priority** | Medium |

Table 42 Description of FR-20

|  |  |
| --- | --- |
| **Identifier** | FR-20 |
| **Title** | View total doctors |
| **Requirement** | If written in a **user perspective**  The admin shall be able to view total number of doctors in a hospital. |
| **Source** | This requirement comes from hospital admin. |
| **Rationale** | To keep an eye on how many doctors hospital has. |
| **Dependencies** | Nil |
| **Priority** | Low |

Table 43 Description of FR-21

|  |  |
| --- | --- |
| **Identifier** | FR-21 |
| **Title** | View total patients |
| **Requirement** | If written in a **user perspective**  The admin shall be able to view total number of patients in a hospital. |
| **Source** | This requirement comes from hospital admin. |
| **Rationale** | To keep an eye on how many patients hospital managed. |
| **Dependencies** | Nil |
| **Priority** | Low |

Table 44 Description of FR-22

|  |  |
| --- | --- |
| **Identifier** | FR-22 |
| **Title** | View total appointments |
| **Requirement** | If written in a **user perspective**  The admin shall be able to view total number of appointments with the doctors in a hospital. |
| **Source** | This requirement comes from hospital admin. |
| **Rationale** | To keep an eye on how many appointments hospital managed. |
| **Dependencies** | Nil |
| **Priority** | Low |

Table 45 Description of FR-23

|  |  |
| --- | --- |
| **Identifier** | FR-23 |
| **Title** | View recent doctors’ information |
| **Requirement** | If written in a **user perspective**  The admin shall be able to view recent doctors list with name, department, contact and status. |
| **Source** | This requirement comes from hospital admin. |
| **Rationale** | To keep an eye on recent doctors information. |
| **Dependencies** | Nil |
| **Priority** | Low |

Table 46 Description of FR-24

|  |  |
| --- | --- |
| **Identifier** | FR-24 |
| **Title** | View recent patients’ info |
| **Requirement** | If written in a **user perspective**  The admin shall be able to view recent patients’ list with name, symptom, contact, address and status. |
| **Source** | This requirement comes from hospital admin. |
| **Rationale** | To keep an eye on recent patients information. |
| **Dependencies** | Nil |
| **Priority** | Low |

Table 47 Description of FR-25

|  |  |
| --- | --- |
| **Identifier** | FR-25 |
| **Title** | View total patients’ appointments |
| **Requirement** | If written in a **user perspective**  The doctor shall be able to view his total number of appointments with his patients. |
| **Source** | This requirement comes from hospital doctor. |
| **Rationale** | For doctor to keep an eye on his appointments. |
| **Dependencies** | Nil |
| **Priority** | Low |

Table 48 Description of FR-26

|  |  |
| --- | --- |
| **Identifier** | FR-26 |
| **Title** | View total doctor’s patients |
| **Requirement** | If written in a **user perspective**  The doctor shall be able to view his total number of his patients. |
| **Source** | This requirement comes from hospital doctor. |
| **Rationale** | For doctor to keep an eye on his patients. |
| **Dependencies** | Nil |
| **Priority** | Low |

Table 49 Description of FR-27

|  |  |
| --- | --- |
| **Identifier** | FR-27 |
| **Title** | View total doctors’ discharged patients |
| **Requirement** | If written in a **user perspective**  The doctor shall be able to view his total number of discharged patients. |
| **Source** | This requirement comes from hospital doctor. |
| **Rationale** | For doctor to keep an eye on his discharged patients. |
| **Dependencies** | Nil |
| **Priority** | Low |

Table 50 Description of FR-28

|  |  |
| --- | --- |
| **Identifier** | FR-28 |
| **Title** | View recent patients’ appointments |
| **Requirement** | If written in a **user perspective**  The doctor shall be able to view his recent appointment list with patient name, picture, description, contact, address and date. |
| **Source** | This requirement comes from hospital doctor. |
| **Rationale** | For doctor to keep an eye on his recent appointment information. |
| **Dependencies** | Nil |
| **Priority** | Low |

Table 51 Description of FR-29

|  |  |
| --- | --- |
| **Identifier** | FR-29 |
| **Title** | View doctors’ patient record |
| **Requirement** | If written in a **user perspective**  The doctor shall be able to view his patients list with patient name, picture, symptoms, contact, address and test. |
| **Source** | This requirement comes from hospital doctor. |
| **Rationale** | For doctor to keep an eye on his recent patient information. |
| **Dependencies** | Nil |
| **Priority** | Low |

Table 52 Description of FR-30

|  |  |
| --- | --- |
| **Identifier** | FR-30 |
| **Title** | View doctors’ discharged patients |
| **Requirement** | If written in a **user perspective**  The doctor shall be able to view his patients list with patient name, admit date, release date, symptoms, contact and address. |
| **Source** | This requirement comes from hospital doctor. |
| **Rationale** | To keep an eye on discharged patients’ information. |
| **Dependencies** | Nil |
| **Priority** | Low |

Table 53 Description of FR-31

|  |  |
| --- | --- |
| **Identifier** | FR-31 |
| **Title** | Fill questionnaire |
| **Requirement** | If written in a **user perspective**  The doctor shall be able to fill the questionnaire with 11 questions displayed to him according to the patient’s checkup.  If written in a **system perspective**  On clicking submit button, the system shall be able to save the values and check from the trained ML model that whether he has PTSD or not. |
| **Source** | This requirement comes from hospital doctor. |
| **Rationale** | To check PTSD in patients. |
| **Business Rule (if required)** | Every question shall be filled with answer, else it will not give results. |
| **Dependencies** | FR-32, FR-33, FR-34 |
| **Priority** | High |

Table 54 Description of FR-32

|  |  |
| --- | --- |
| **Identifier** | FR-32 |
| **Title** | Get results |
| **Requirement** | If written in a **user perspective**  The doctor shall be able to view the results obtained.  If written in a **system perspective**  On clicking submit button, the system shall give two types of results.   1. PTSD detected   On detection, the doctor has to click on Start button to add the advice to the patient.   1. PTSD not detected |
| **Source** | This requirement comes from hospital doctor. |
| **Rationale** | To check PTSD in patients. |
| **Dependencies** | FR-33, FR-34 |
| **Priority** | High |

Table 55 Description of FR-33

|  |  |
| --- | --- |
| **Identifier** | FR-33 |
| **Title** | Add prescriptions |
| **Requirement** | If written in a **user perspective**  The doctor shall be able to add the recommended test, advice and prescription for the patient. |
| **Source** | This requirement comes from hospital doctor. |
| **Rationale** | To advice the patient if PTSD is detected. |
| **Dependencies** | FR-34 |
| **Priority** | Medium |

Table 56 Description of FR-34

|  |  |
| --- | --- |
| **Identifier** | FR-34 |
| **Title** | Download prescription |
| **Requirement** | If written in a **user perspective**  The doctor shall be able to download the advice as a pdf and send to the patient.  If written in a **system perspective**  After clicking the download option, the system shall download the advice with date to the PC. |
| **Source** | This requirement comes from hospital doctor. |
| **Rationale** | To save the record and send to the patient. |
| **Dependencies** | Nil |
| **Priority** | Medium |

Table 57 Description of FR-35

|  |  |
| --- | --- |
| **Identifier** | FR-35 |
| **Title** | View patients’ appointment |
| **Requirement** | If written in a **user perspective**  The doctor shall be able to view his patients’ appointment list with \_\_? |
| **Source** | This requirement comes from hospital doctor. |
| **Rationale** | Doctor to keep an eye on his appointments details. |
| **Dependencies** | Nil |
| **Priority** | Low |

Table 58 Description of FR-36

|  |  |
| --- | --- |
| **Identifier** | FR-36 |
| **Title** | Delete patient’s appointment |
| **Requirement** | If written in a **user perspective**  The doctor shall be able to delete his patient’s appointment.  If written in a **system perspective**  On clicking delete, the system shall remove the appointment from admin and doctors dashboard. |
| **Source** | This requirement comes from hospital doctor. |
| **Rationale** | For doctor if he wants to remove any appointment. |
| **Dependencies** | Nil |
| **Priority** | Low |

Table 59 Description of FR-37

|  |  |
| --- | --- |
| **Identifier** | FR-37 |
| **Title** | View doctor’s info |
| **Requirement** | If written in a **user perspective**  The patient shall be able to view doctor’s name, symptoms, contact, address and department of the doctor he is being appointed to. |
| **Source** | This requirement comes from patient. |
| **Rationale** | For patient to know his doctor’s details. |
| **Dependencies** | FR-2, FR-18, FR-19 |
| **Priority** | Medium |

Table 60 Description of FR-38

|  |  |
| --- | --- |
| **Identifier** | FR-38 |
| **Title** | View admit date |
| **Requirement** | If written in a **user perspective**  The patient shall be able to view the day he is being admitted to hospital. |
| **Source** | This requirement comes from patient. |
| **Rationale** | For patient to know his admit date. |
| **Dependencies** | FR-18, FR-19 |
| **Priority** | Medium |

Table 61 Description of FR-39

|  |  |
| --- | --- |
| **Identifier** | FR-39 |
| **Title** | View his appointment (patient) |
| **Requirement** | If written in a **user perspective**  The patient shall be able to view his doctor’s name, description, date and status of his appointment. |
| **Source** | This requirement comes from patient. |
| **Rationale** | For patient to know his appointment details. |
| **Dependencies** | FR-18, FR-19 |
| **Priority** | Medium |

Table 62 Description of FR-40

|  |  |
| --- | --- |
| **Identifier** | FR-40 |
| **Title** | View discharge bill |
| **Requirement** | If written in a **user perspective**  The admin shall be able to view the bill he has generated.  The patient should able to view the discharge bill, admin has saved in the system. |
| **Source** | This requirement comes from patient, admin. |
| **Rationale** | For patient to pay the discharge bill.  For admin to review the bill. |
| **Dependencies** | FR-14, FR-15 |
| **Priority** | High |

Table 63 Description of FR-41

|  |  |
| --- | --- |
| **Identifier** | FR-41 |
| **Title** | Logout |
| **Requirement** | If written in a **user perspective**  The admin, doctor and patient shall be able to logout from the system.  If written in a **system perspective**  After clicking logout button, system shall take admin, doctor and patient to the main display page. |
| **Source** | This requirement comes from patient, doctor and admin. |
| **Rationale** | For admin, doctor and patient to logout whenever the web application is not in use. |
| **Dependencies** | Nil |
| **Priority** | High |

## Non-Functional Requirements

### Reliability

As algorithm is pre-developed, the application is likely to not fail. No definition of failure is currently known and no mean time between failures is recognized.

**SRS01 Availability**

### Usability

The requirements for usability involve accessibility, mistake avoidance and recovery, interaction efficiency, and simplicity of learning and use. The user interface designer may produce the best user experience by following these usability guidelines.

*USE-1: The PTSD detection platform allows a patient to access his appointed doctor.*

*USE-2: The web page provides a uniform look and feel between all the web pages.*

*USE-3: The web page provides a user-friendly interface.*

*USE-4: Database will maintain all of the information even of the patients who are discharged patients.*

*USE-5: The admin would be able to approve new patients and doctors.*

### Performance

**SRS02 Response Time**

After verifying the patient's details, system must respond in 1 sec.

**SRS 03 Capacity**

1000 persons at once must be supported by system.

**SRS04 User-interface**

Within 5 seconds, user interface screen must reply.

**SRS05 Conformity**

Applications must follow Microsoft Accessibility recommendation.

### Security

The system is being built on Django Framework and SQLite database. It will have all the built-in security and protection provided by both the framework and database. No external security, such as physical firewalls or software protection is being used in the system.

**STS06 Patient identification**

The system demands that the user log in using their login information.

**SRS07 Login ID**

Every system user must have a login ID and password.

**SRS08 Admin in the hospital**

The hospital's database's administrator will be able to access all of the data there, add new patients and doctors, and alter any of the data as well. Any database modification (insert, remove, or update) must be synced as well as carried out by the administrator alone.

**SRS09 Administrators' Rights**

Administrators shall be able to view as well as modify all information in hospital’s database

## External Interface Requirements

Information is provided in this part to make sure that users and other hardware or software components may interface with the system effectively. There should be a distinct interface specification or system architecture specification for a complicated system with several subcomponents. Material from other papers may be referenced in the interface documentation. For instance, it might include a link to a handbook for a hardware device that describes the error messages that the hardware is capable of sending to software.

### User Interfaces Requirements

Post-Traumatic Stress Disorder (PTSD) Prediction and consultation system uses a web-based platform mainly a customized application which is accessed by only authorized users to detect the presence of trauma (Group, et al., 2021). This helps in proceeding towards the cure of PTSD.

The user interface is a fundamental part of every software, hardware, or hybrid system (UI). The ease of use of the product is the main factor influencing user adoption. Only the user interface of this system may be understood by users. The following are our application's user interface and logistical features.

#### Simple interface

The user experience for our program is straightforward and engaging. The software doesn't use a lot of graphics, and each screen has a consistent user interface (UI) with carefully placed text views, edit messages, buttons, photos, and icons.

#### Display

Hospital management system, PTSD detected in patients or not, Patient Record, Doctor Record, symptoms, Prescription, Medicine, and Tests.

#### Buttons

Login, signup, view admin, view doctor, view patient, view appointments, take appointment, recent doctors, recent patients, total doctors, total patients, total appointments, doctor record, register doctor, approve doctor, doctor specialization, patient record, admit patient, approve patient, discharge patient, book appointment, approve appointment, delete appointment, generate fee bill, download fee bill, start, get result and download .

Every screen has buttons, each having a distinct function and purpose. Every button has a function and may be clicked. After confirming that all of the given information is accurate and in the right format, clicking the Signup button registers the user.

The login button verifies in the database that the user has provided the right credentials before allowing access to the application.

### Software interfaces

The software interface can be the application which is used to interact with users which can be a doctor or hospital administration. The tools which can be used for the development of this application in prediction of PTSD are Visual Studio Code. We can download them from the internet for research and development. Many of them may be free and many may not. We use the one which would be best suitable for us and easy to understand because this is a completely new thing which we are going to explore more and more for its better understanding. In case of a subscription plan, we request the university to help us with it or we can manage it by ourselves too.

SI-1: *PTSD Predictor*

### Hardware interfaces

We are building up Software, there are no such Hardware Interface requirements for our project.

The only Hardware we’ll need is:

1. A GPU required training the deep learning model. Google CoLab or a Physical GPU that supports CUDA (minimum Nvidia GTX 1050 ti) is required.
2. Around 10iGB Storage to store dataset, required for training and displaying on the website. A scalable database storage that expands as the user base grows.
3. We use a laptop, personal computer or similar device for this application. Device should have a RAM of 8GB, an SSD card for better performance and speed, and if anyone has a GTX NVIDIA graphics card then it’s also beneficial.
4. All of these components are available in the market. There are different prices but we can also get this type of laptop for about Rs.90000 minimum but its maximum price can go up to 300, 000

### Communications interfaces

A good quality internet is a basic requirement which helps to run the application. The communication standards used for this application are HTTP, FTP and SMTP.

*CI-1: The Communication Interface we are using will check if user is registered or not. Only registered and authorized users will have access to web application.*

*This will ensure no third person is trying to register as approval will be required from admin.*

# Design and Architecture



## Architectural Design

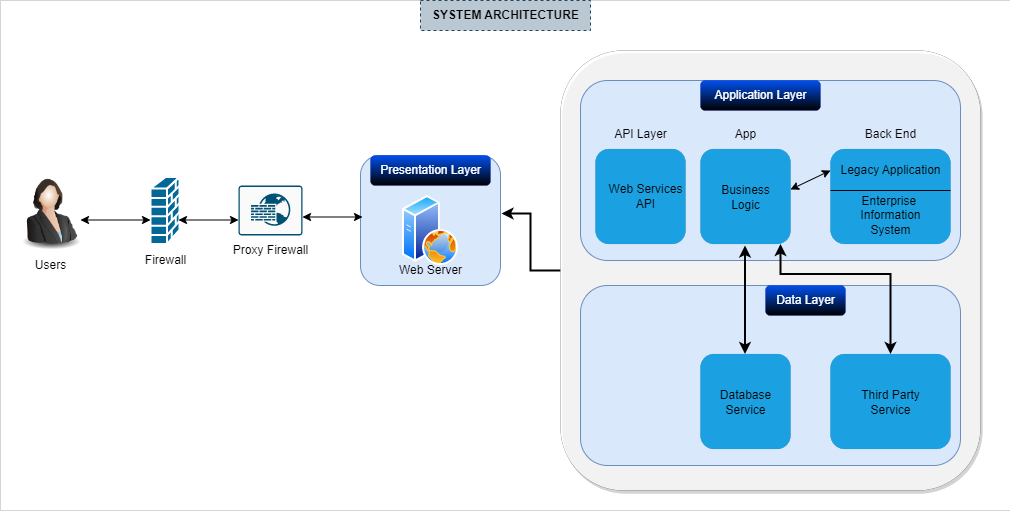
* In initial design stage create Box and Line Diagram for simpler representation of the systems

Figure 2 Box and Line Diagram



* After finalizing architecture style/pattern diagram (MVC, Client-Server, Layered, Multi-tiered) we have created a detailed mapping modules/component to each part of the architecture

Figure 3 Architecture Diagram



## Design Models

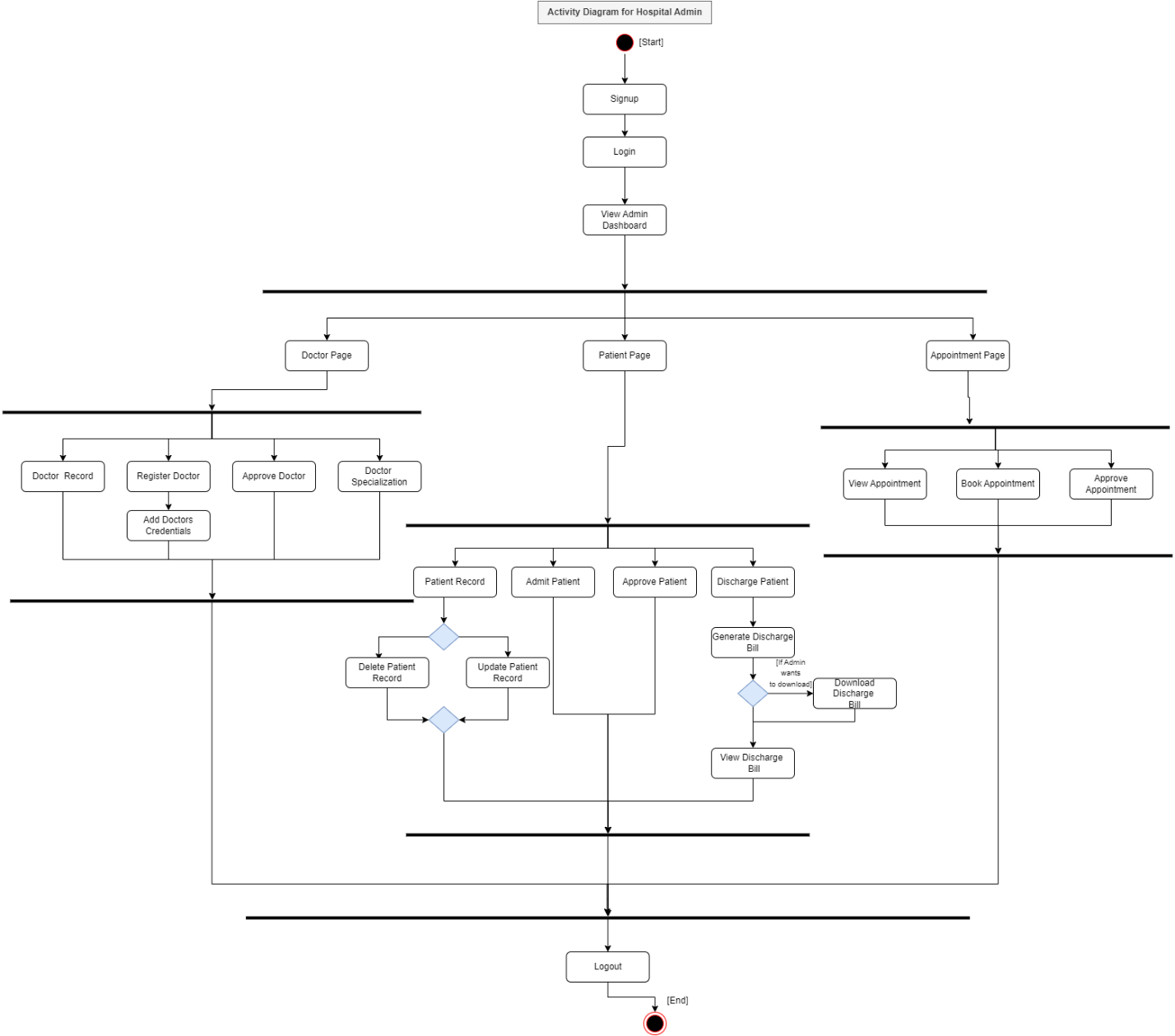
We have created design models as are applicable to our system.

***Design Models for Object Oriented Development Approach***

We have used object-oriented development approach in our project which includes:

* Activity Diagram

Figure 4 Activity Diagram for Admin



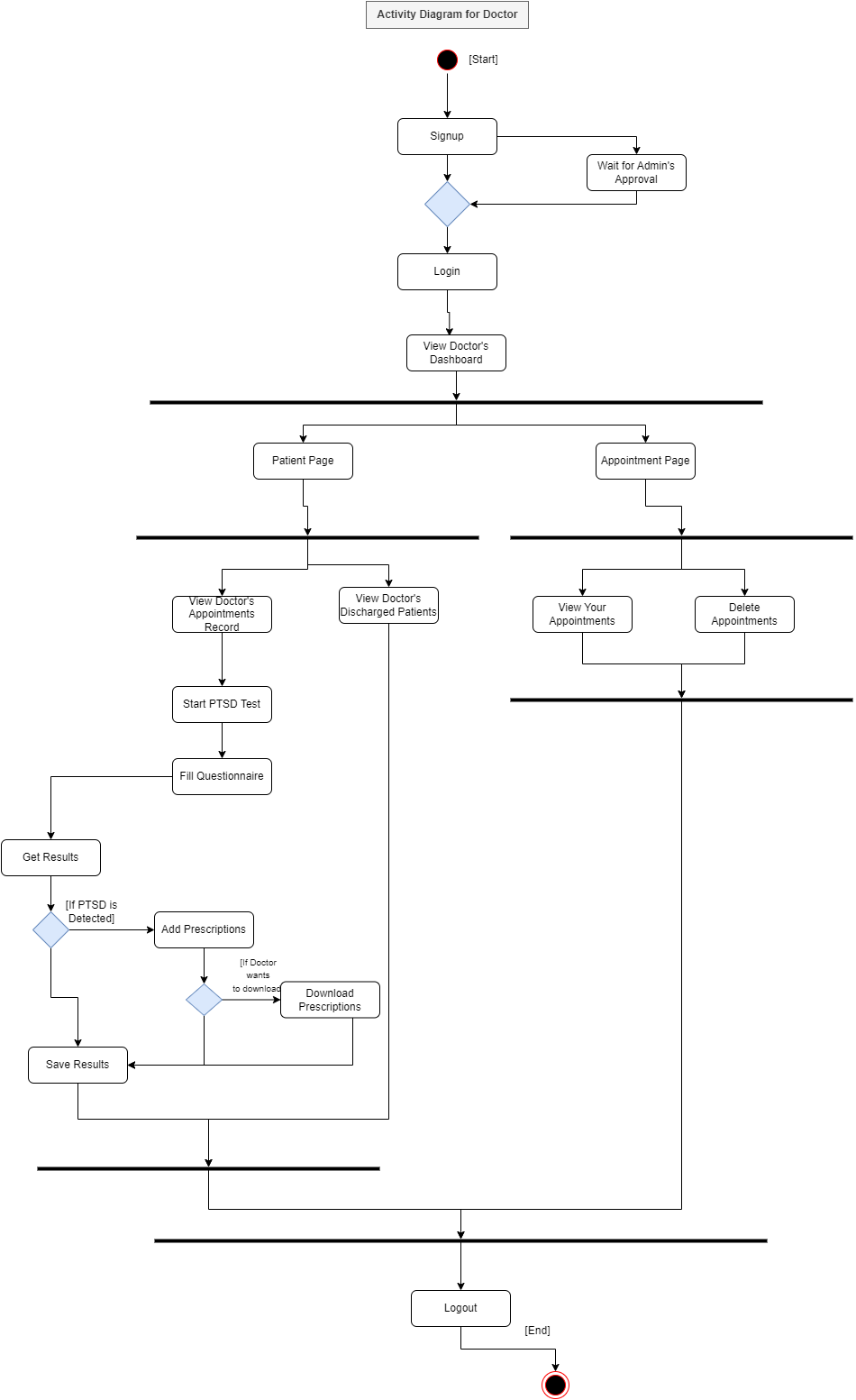
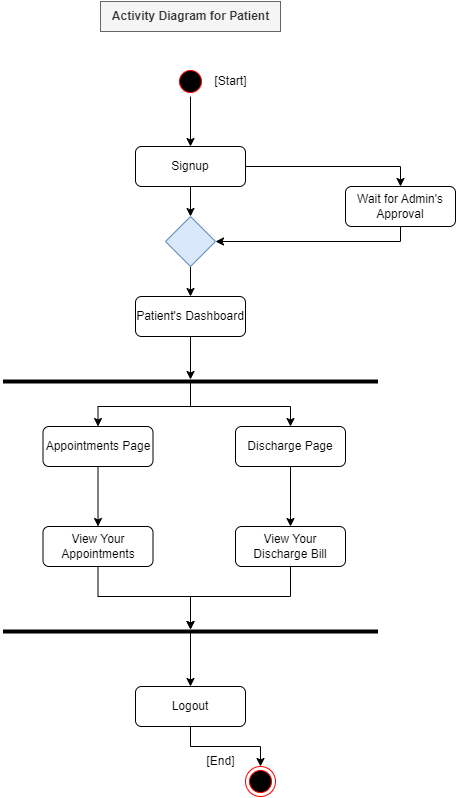
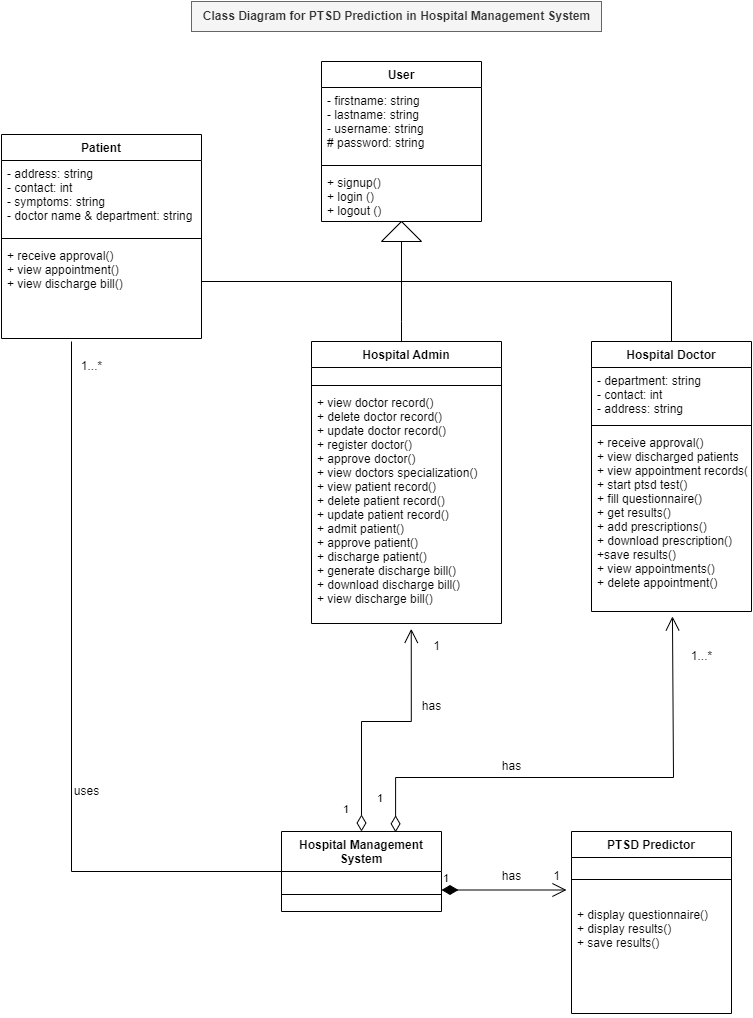


Figure 5 Activity Diagram for Doctor

Figure 6 Activity Diagram for Patient

* Class Diagram

Figure 7 Class Diagram

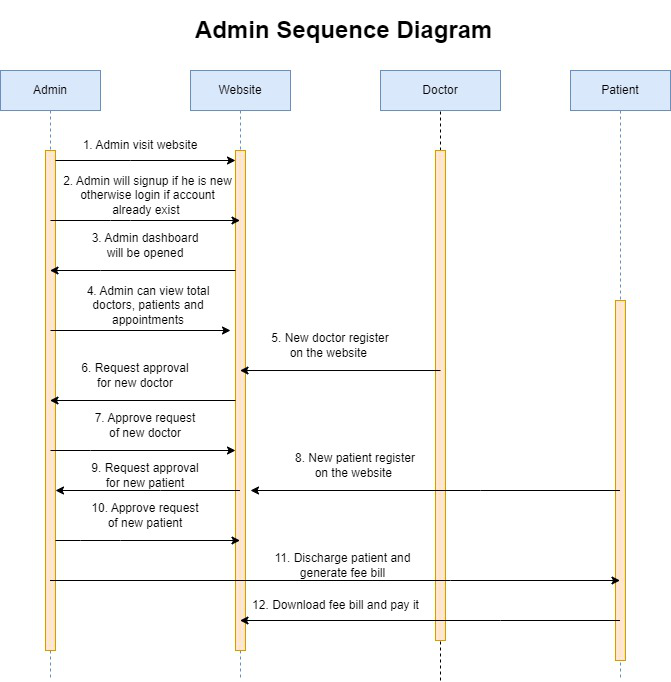
* Sequence Diagram

Figure 8 Admin Sequence Diagram

Figure 9 Doctor Sequence Diagram

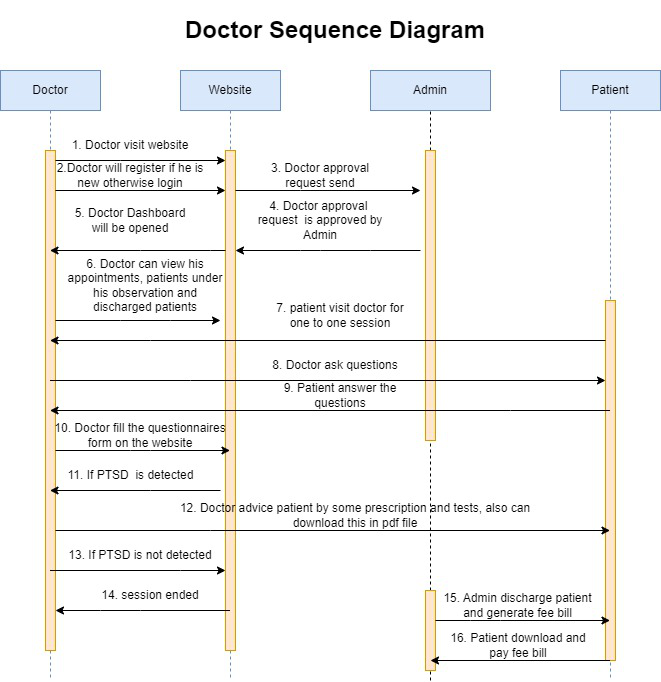
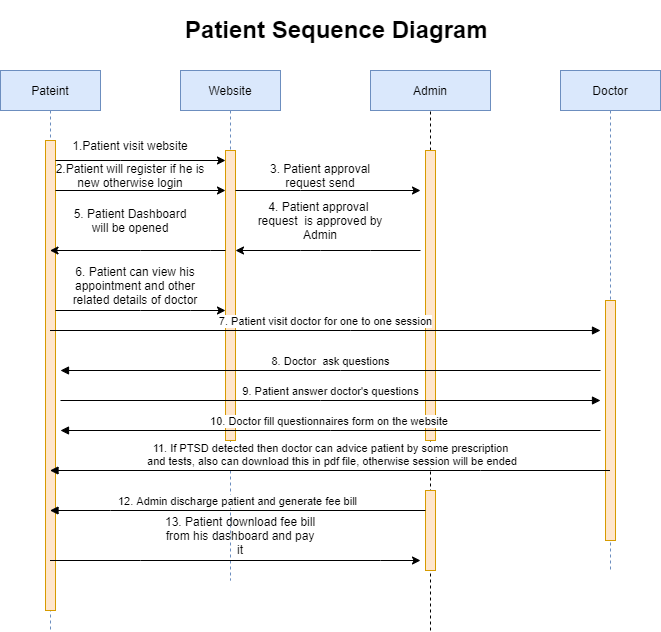


Figure 10 Patient Sequence Diagram



* State Transition Diagram (for the projects which include event handling and backend processes)

Figure 11 Admin State Transition Diagram

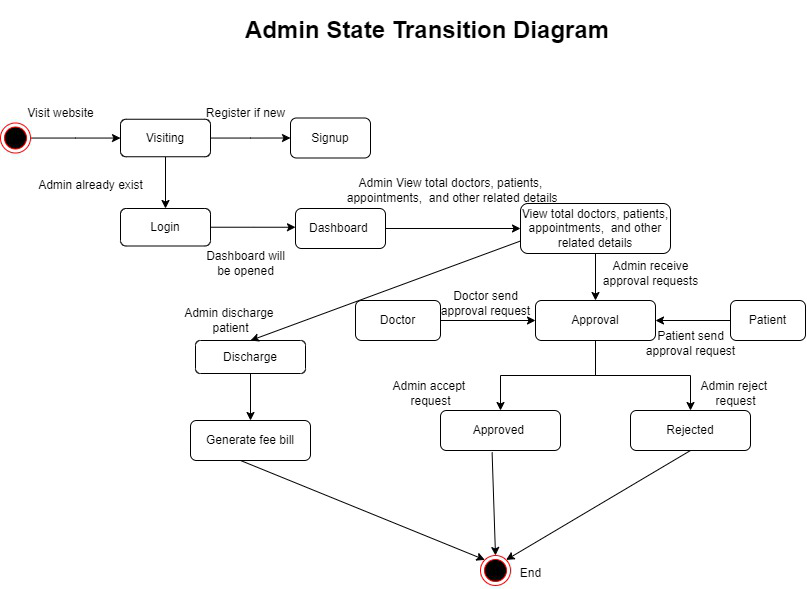
******

Figure 12 Doctor State Transition Diagram

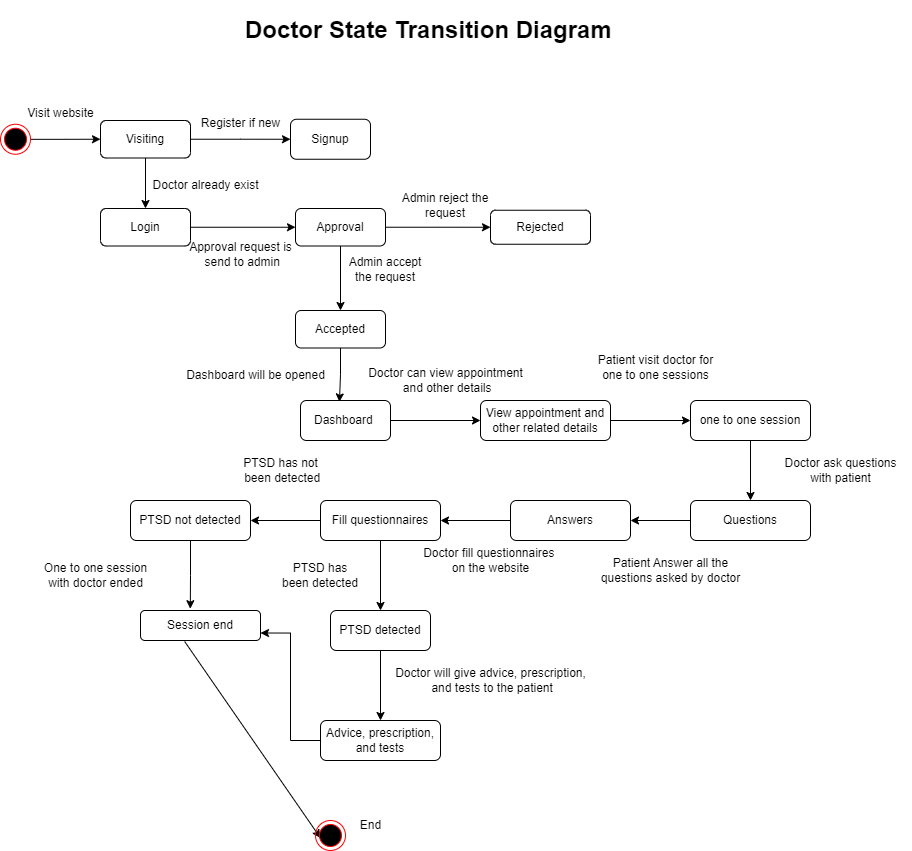
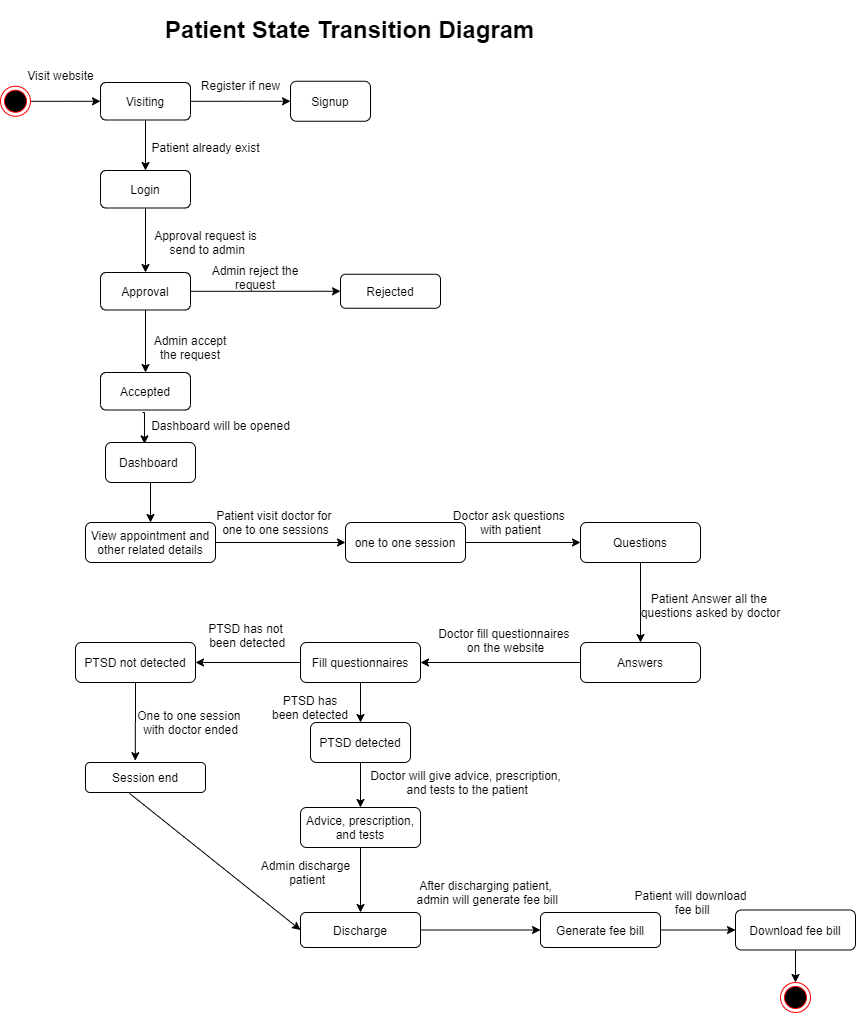
******

Figure 13 Patient State Transition Diagram

******

## Data Design

The information related to all the doctors, patients, admin, appointments, login and signup credentials, generated fee bill, generated advice file, approved users, discharged patients, and PTSD questionnaires form are all stored in backend into sqlite3 database which is integrated with our website (Foundation, 2023). Additionally features after filling questionnaires form will be updated in .csv file.

Login and signup credentials will be cross checked from database that whether they are accurate or the user already exist or he is new on this website. Passwords are stored in hashed format in database. When doctor will fill questionnaires form then all the features will be compared and processed from dataset which is present in .csv file integrated with our website. ML training was done on Jupyter notebook and then it was embedded in PTSD prediction module in which doctor fill questionnaires form by one to one session with patient and if PTSD would be detected then doctor will advise, specify some tests and prescription which patient have to take else if PTSD is not detected session will end and admin will discharge patient and generate fee bill which would be send to patient dashboard so that he can download it and pay it. Every new doctor and patient have to take approval from admin to access his dashboard this would ensure the security of the system.

### Data Dictionary

**Table 64 Data Dictionary**

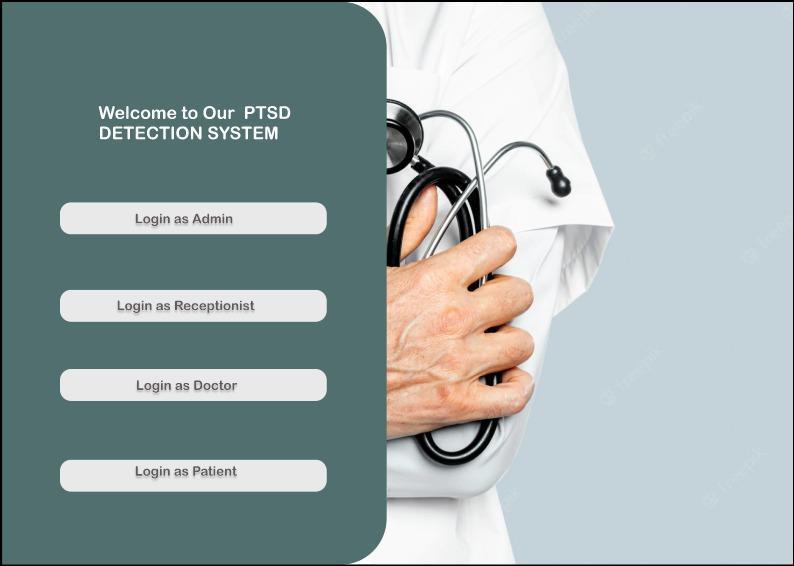
|  |  |
| --- | --- |
| **System Entities** | **Types & Descriptions** |
| hospital\_patient | id(integer not null primary key autoincrement), address(varchar(40) not null), mobile(varchar(20) null), symptoms(varchar(100) not null), status(bool not null), user\_id(integer not null references “auth\_user”(“id”)), admitDate(date not null), profile\_pic(varchar(100)), assignedDoctorId(integer unsigned) |
| hospital\_doctor | Id(integer not null primary key autoincrement), address(varchar(40) not null), mobile(varchar(20)), status(bool not null), user\_id(integer not null references “auth\_user” (“id”)), profile\_pic(varchar(100)), department(varchar(50) not null) |
| hospital\_appointment | Id(integer not null primary key autoincrement), doctorId(integer unsigned), appointmentDate(date not null), description(text not null), status(bool not null), doctorName(varchar(40)), patientName(varchar(40)), patientId(integer unsigned) |
| hospital\_diagnosis | Id(integer not null primary key autoincrement), patientId(integer unsigned), test(text not null), doctorsAdvice(text not null), prescription(text not null) |
| hospital\_patientdischargedetails | Id(integer not null primary key autoincrement), patientName(varchar(40) not null), assignedDoctorName(varchar(40) not null), address(varchar(40) not null), mobile(varchar(20)), symptoms(varchar(100)), admitDate(date not null), releaseDate(date not null), daySpent(integer unsigned not null), roomCharge(integer unsigned not null), medicineCost(integer unsigned not null), doctorFee(integer unsigned not null), OtherCharge(integer unsigned not null), total(integer unsigned not null), patientId(integer unsigned) |
| auth\_group | Id(integer not null primary key autoincrement), name(varchar(150) not null) |
| auth\_group\_permissions | Id(integer not null primary key), group\_id(integer not null references “auth\_group” (“id”)), permission\_id(integer not null references “auth\_permission” (“id”)) |
| auth\_permission | Id(integer not null primary key autoincrement), content\_type\_id(integer not null references “django\_content\_type” (“id”)), codename(varchar(100) not null), name(varchar(255) not null) |
| auth\_user | Id(integer not null primary key autoincrement), password(varchar(128) not null), last\_login(datetime), is\_superuser(bool not null), username(varchar(150) not null), first\_name(varchar(30) not null), email(varchar(254) not null), is\_staff(bool not null), is\_active(bool not null), date\_joined(datetime not null), last\_name(varchar(150) not null) |
| auth\_user\_groups | Id(integer not null primary key autoincrement), user\_id(integer not null references “auth\_user” (“id”)), group\_id(integer not null references “auth\_group” (“id”)) |
| auth\_user\_user\_permissions | Id(integer not null primary key ), user\_id(integer not null references “auth\_user” (“id”)), permission\_id(integer not null references “auth\_permission” (“id”)) |
| django\_admin\_log | Id(integer not null primary key autoincrement), action\_time(datetime not null), object\_id(text), object\_repr(varchar(200) not null), change\_message( text not null), content\_type\_id(integer references “django\_content\_type” (“id”)), user\_id(integer not null references “auth\_user” (“id”)), action\_flag(smallint unsigned not null) |
| django\_content\_type | Id(integer not null primary key autoincrement), app\_label( varchar(100) not null), model(varchar(100) not null) |
| django\_migrations | Id(integer not null primary key autoincrement), app(varchar(255) not null), name(varchar(255) not null), applied(datetime not null) |
| django\_session | session\_key(varchar(40) not null primary key), session\_data(text not null), expire\_date(datetime not null) |

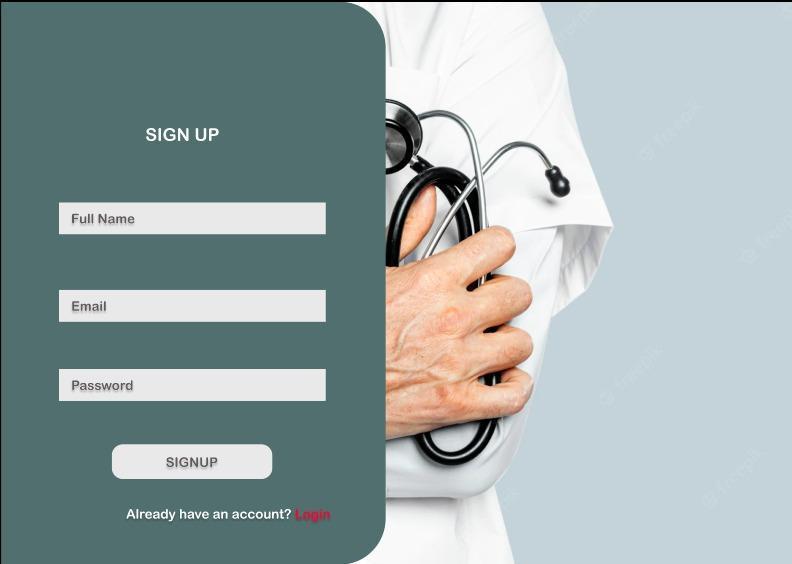
## Human Interface Design

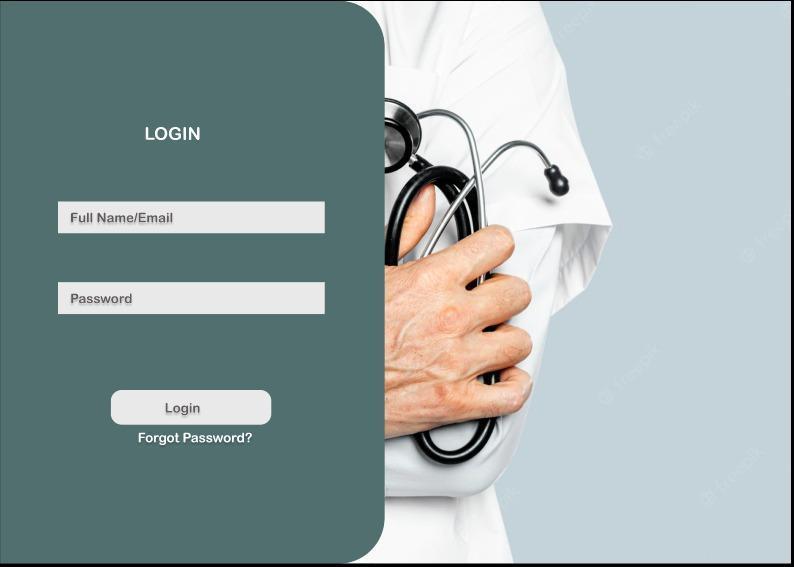
From the viewpoint of the user, we have described how the system functions. We have shown how the user can utilize our system to fulfill all the anticipated features as well as the feedback data that will be shown to them.

### Screen Images

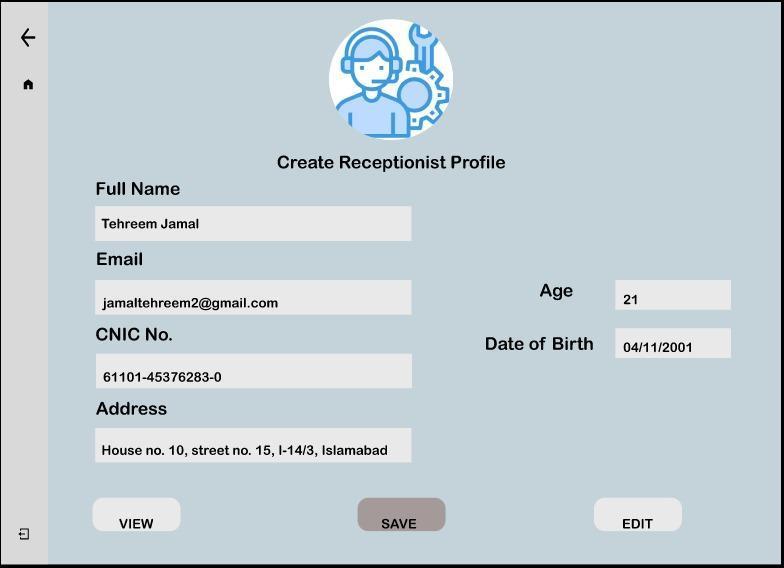
This is our rough draft of User Interface before we started our work. When we started developing and by adding ideas of our respected teachers, the project became better. So, this Interface design is basically what we planned and in User Interface Design, we have put our web platform UI of what we designed.

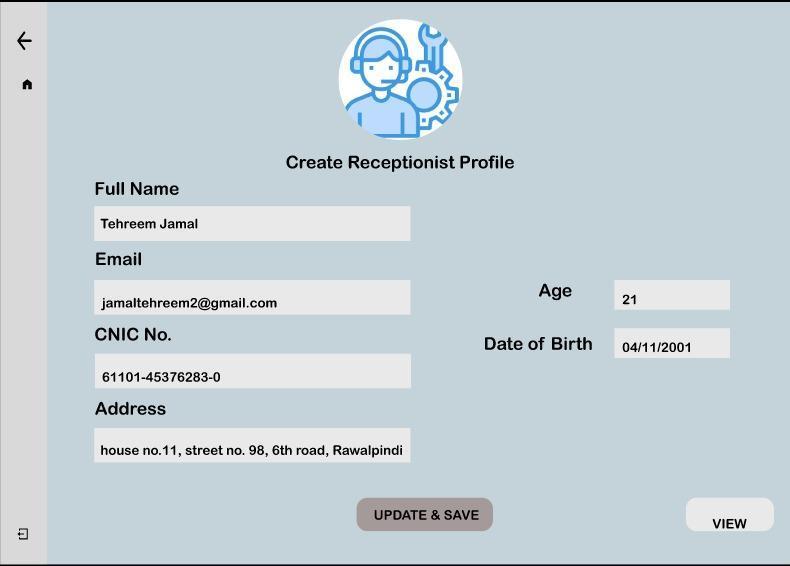
Signup page for Admin, Receptionist and Doctor



Login page for Admin, Receptionist, Patient and Doctor

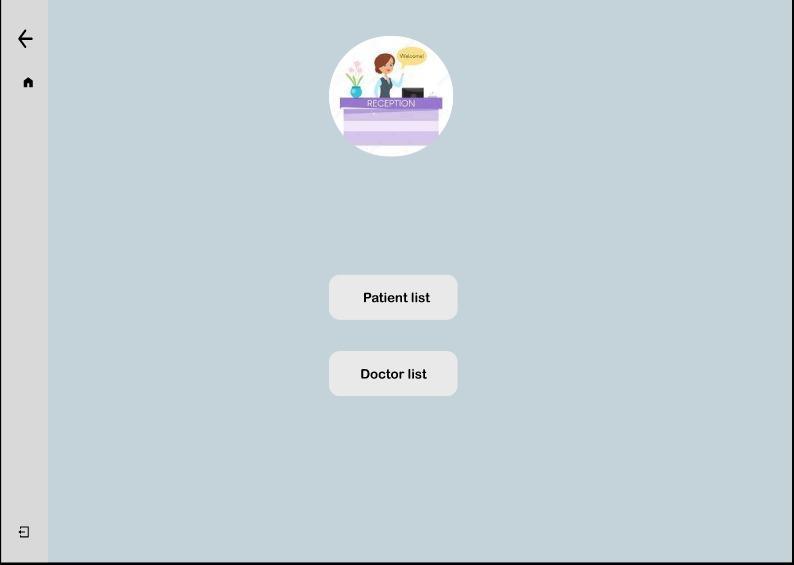
**Login as Admin:**

He will create Receptionists profile

He is updating Receptionist’s profile

**Login as Receptionist:**

He can search Patient and search Doctor.

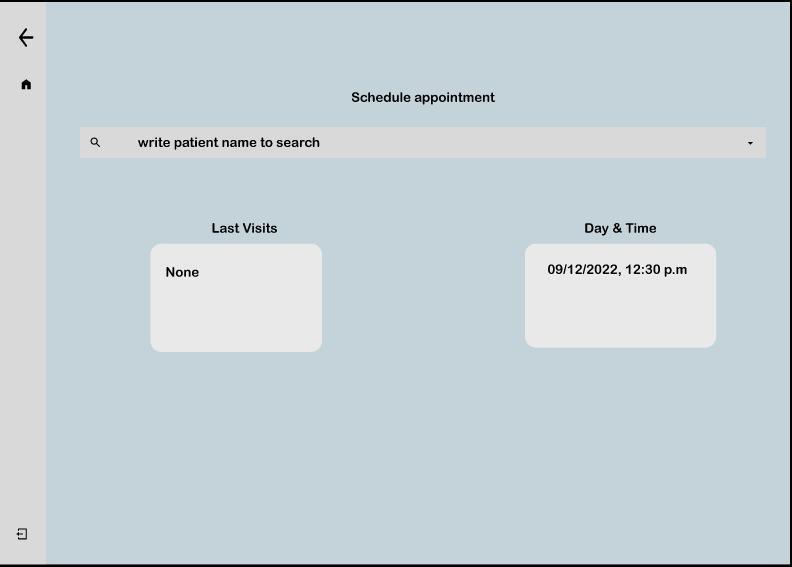


On clicking Patient List:

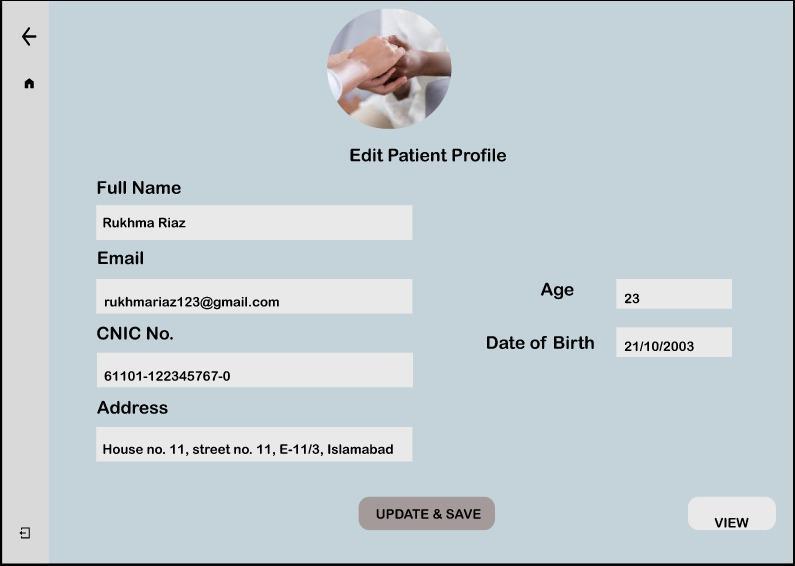


On clicking Add patient,

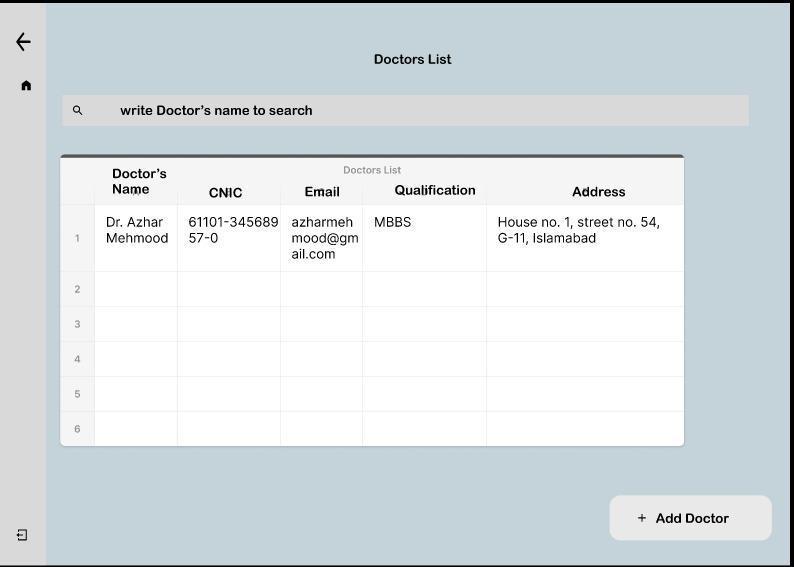
On clicking schedule appointment,



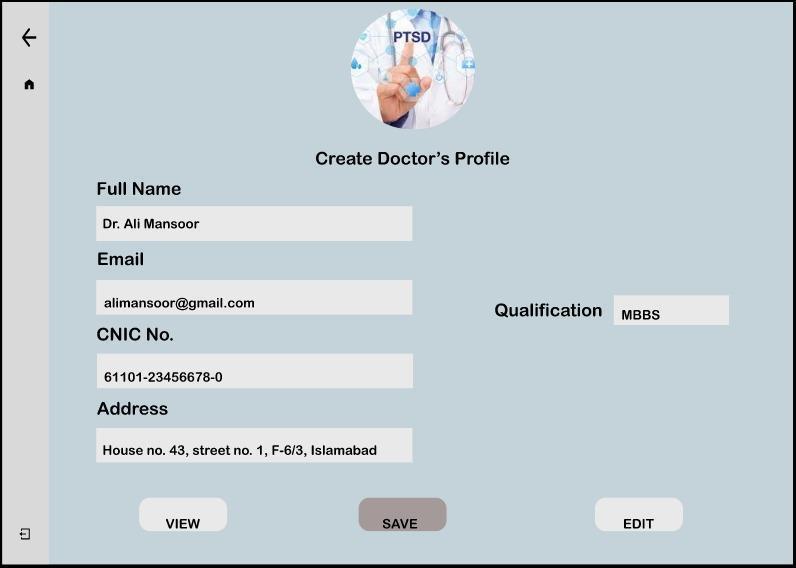
On clicking, Edit button in “Add Patient” form to update the information.



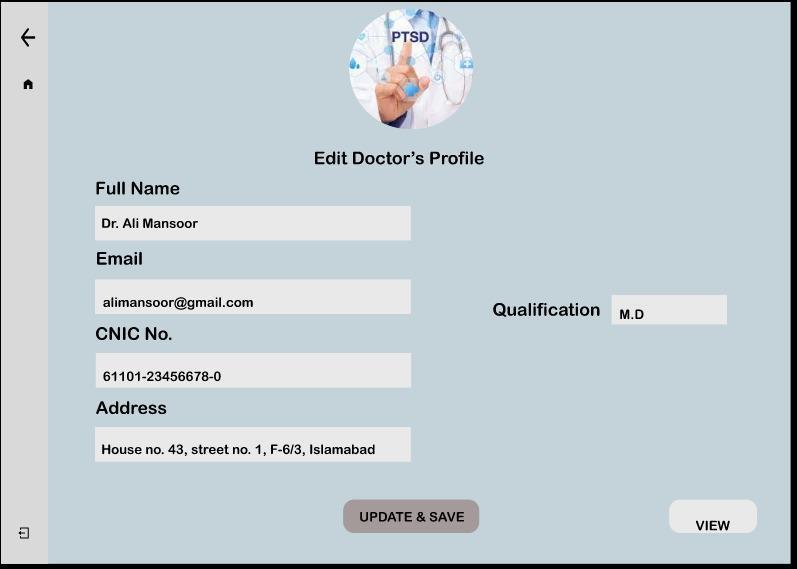
Search the Doctor, If not present then, Add Doctor.



On clicking “Add Doctor” button, form is filled.

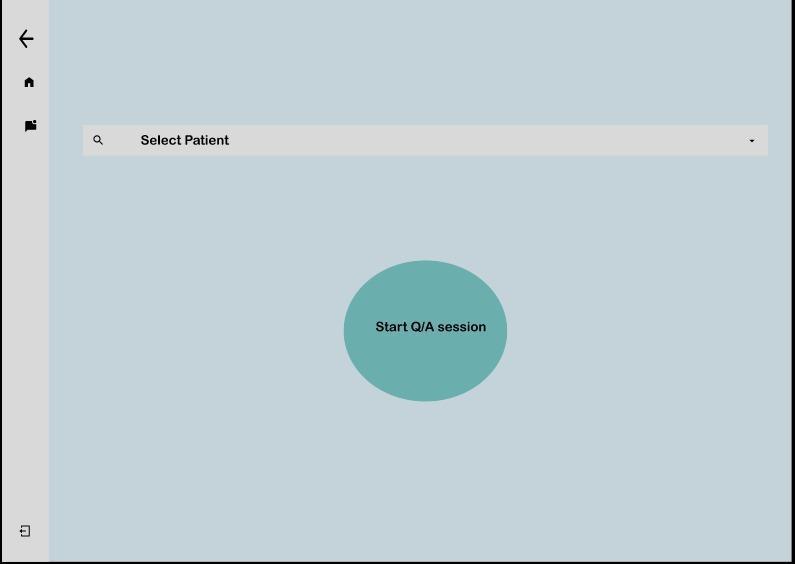


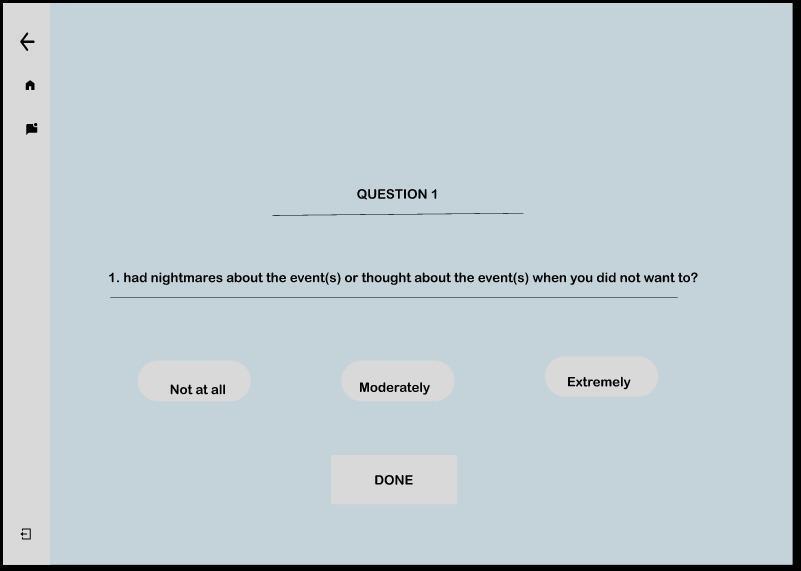
On clicking, Edit button, Qualification of a Doctor is updated.

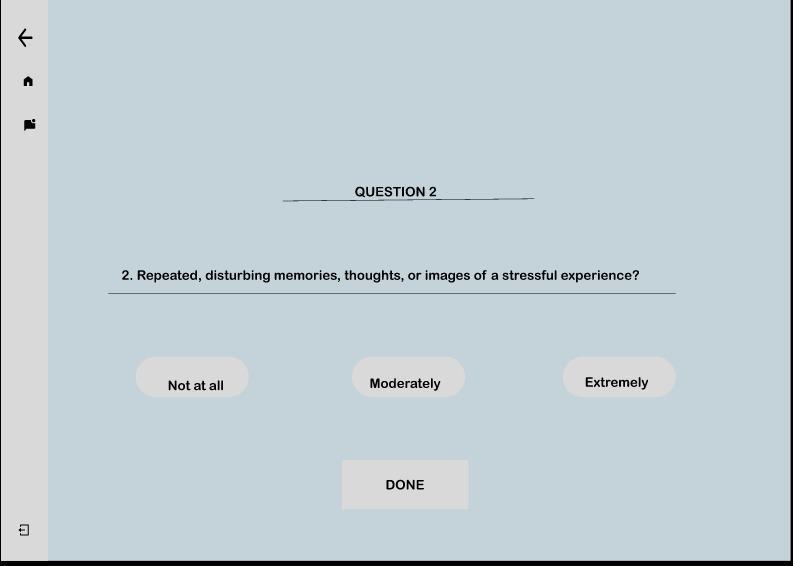


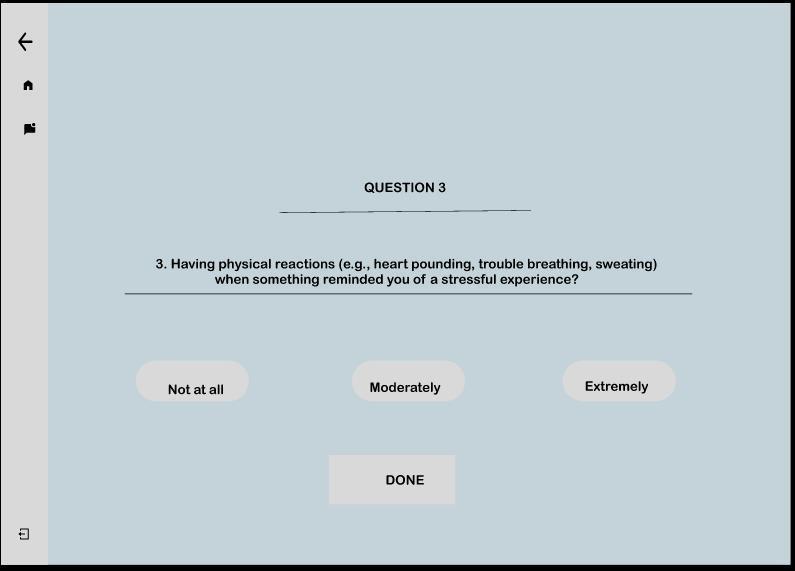
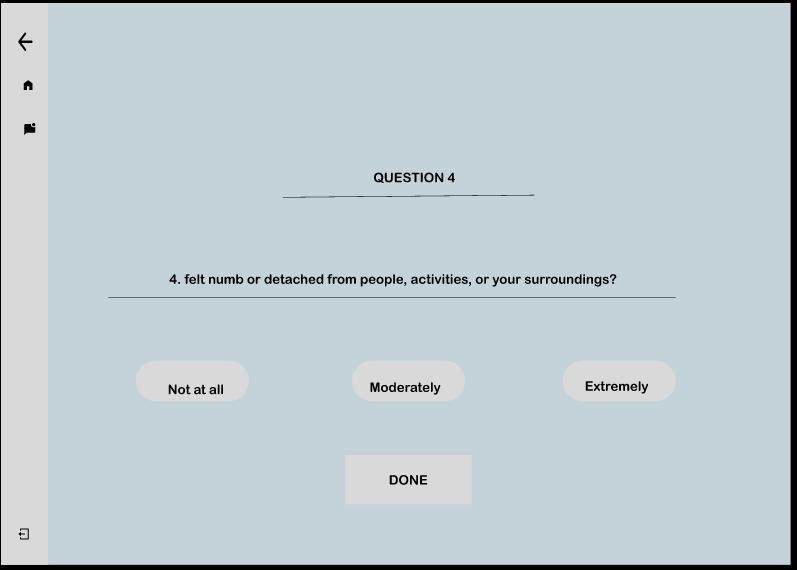
**Login as Doctor:**

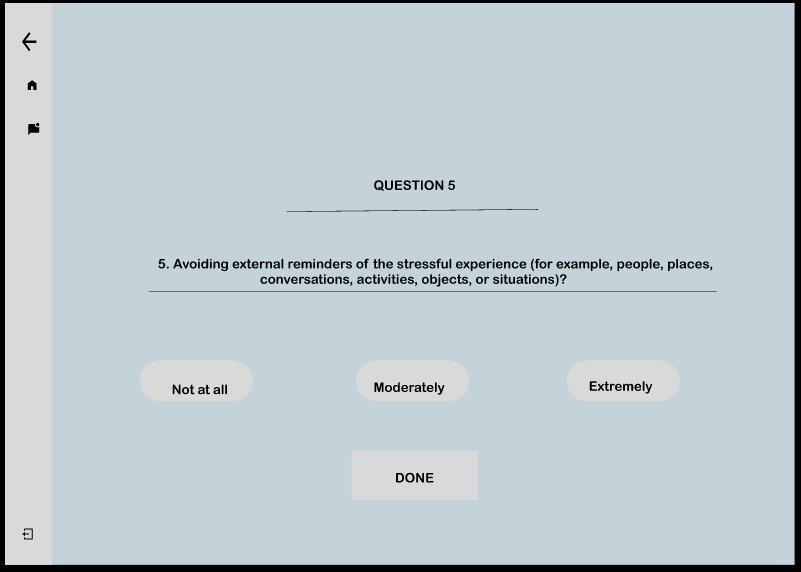
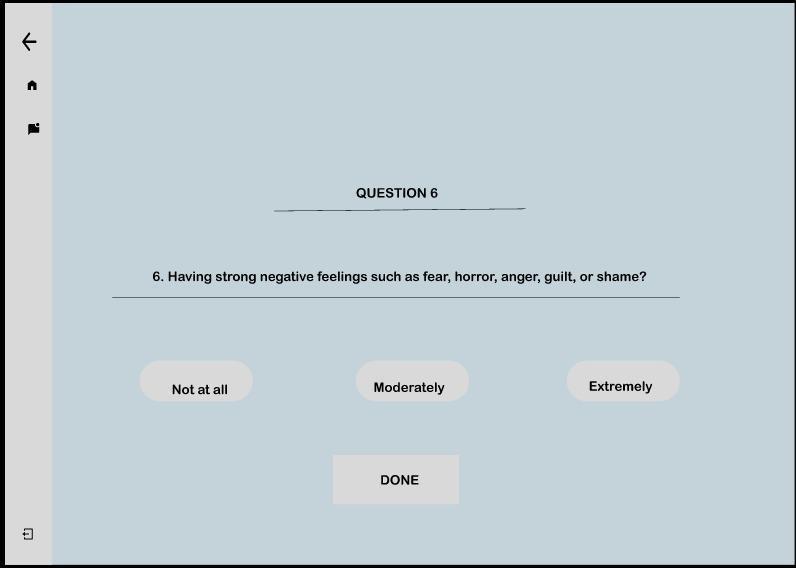
Search the name of Patient and Start Question Answer session.

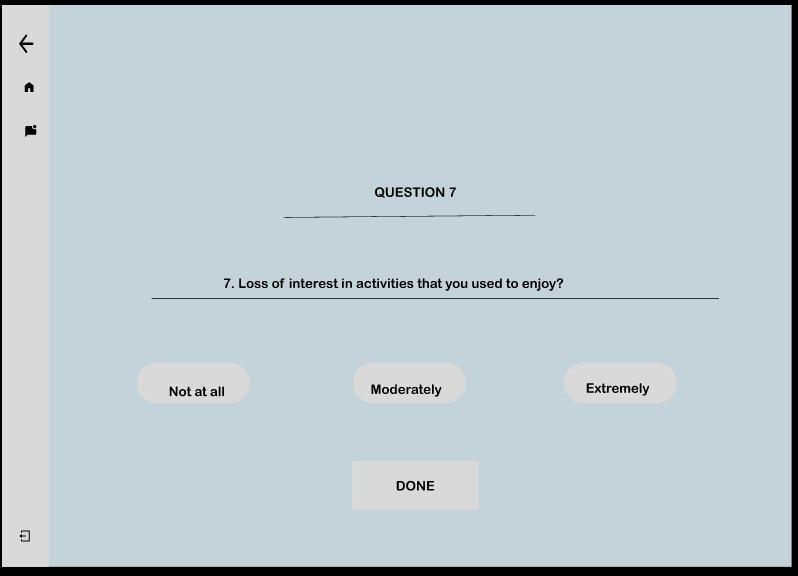
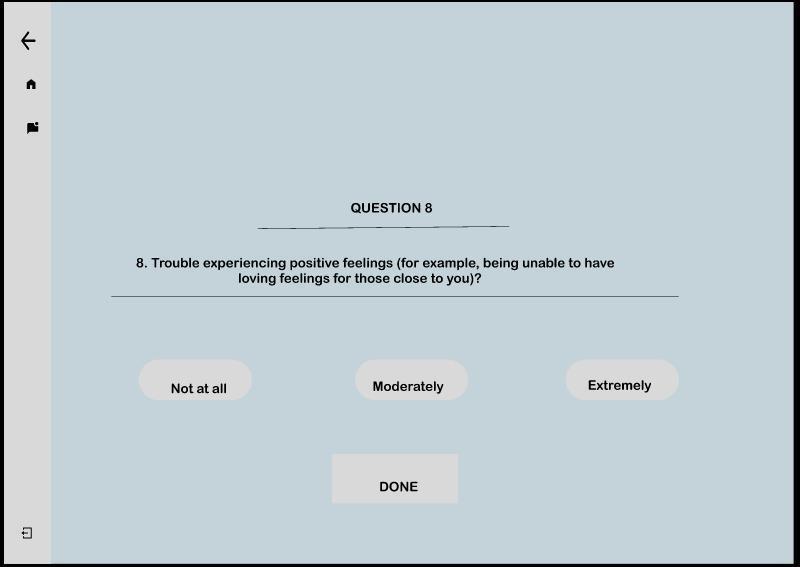


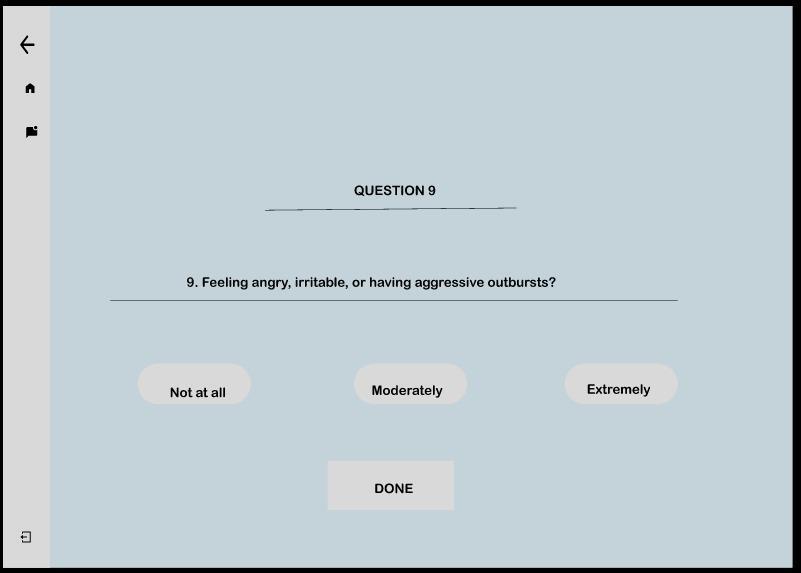


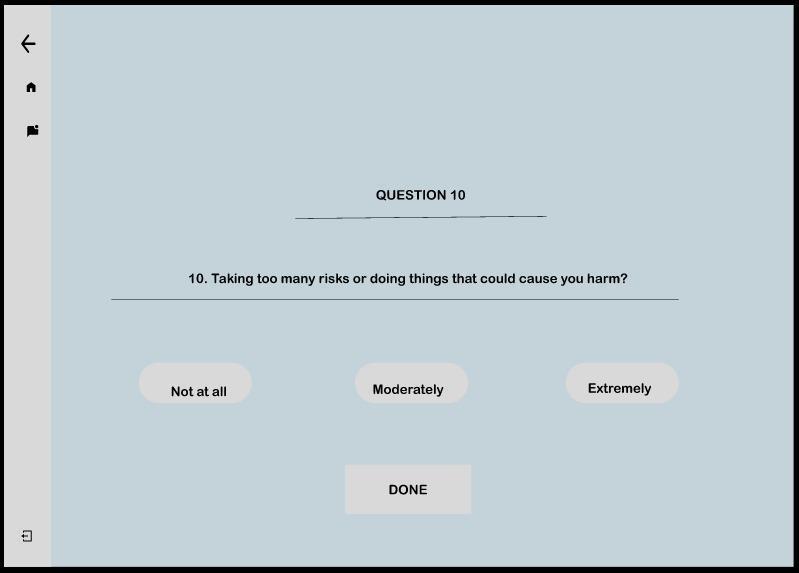


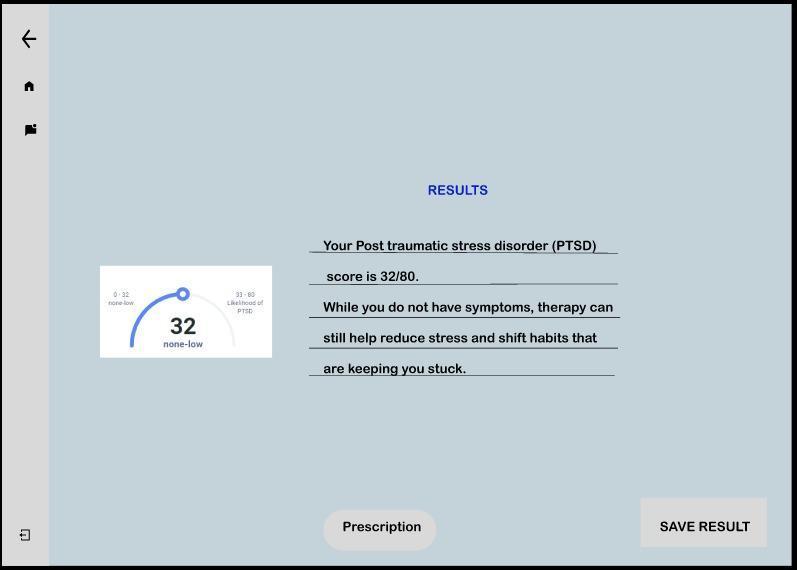




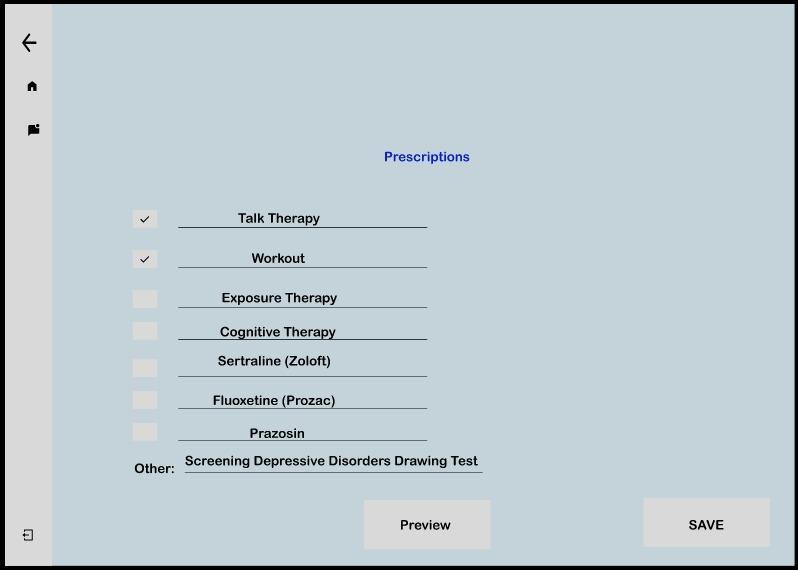


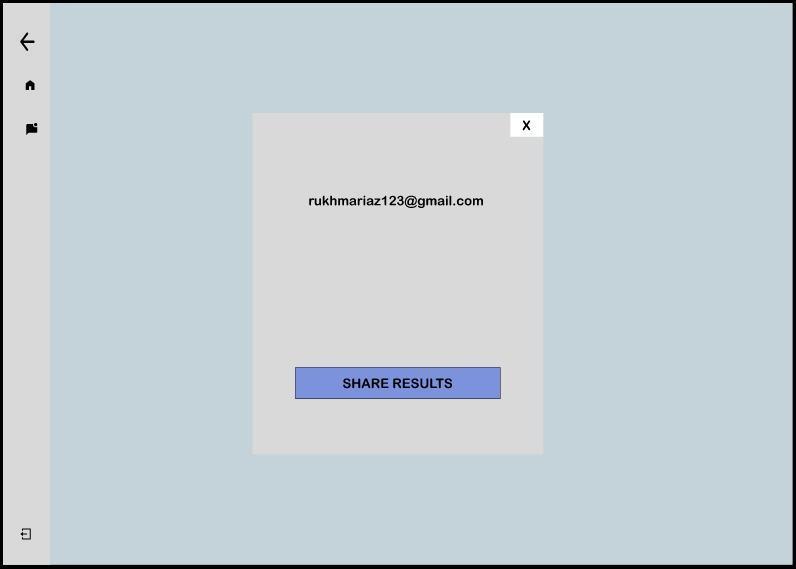




Results shown through ML model implementation in the backend.

Select Prescriptions in which advice, medicines and tests are included



Results of the patient is shared with him/her via email.

If patient is facing any issue, he/she can text the Doctor via Chatbot.



Logout option for all Admin, Receptionist, Doctor and Patient.

### Screen Objects and Actions

A discussion of screen objects as well as actions associated with those objects are as follows:

1. Signup 🡪box will appear where doctor, patient and admin have to enter its credentials.
2. Login 🡪admin, doctor and patient to enter their respective dashboards.

**ADMIN:**

1. Doctor page 🡪 four boxes below will appear.

* Doctor record 🡪 displays all doctors record in table.
* Register doctor 🡪 admin can register the doctor.
* Approve doctor 🡪 admin will approve the doctor’s registration.
* Doctor specialization 🡪 admin can view Doctor specialization

1. Patient page 🡪 four boxes below will appear.

* Patient record 🡪 displays all patients’ record in a table.
* Admit patient 🡪 admit the patient to a hospital.
* Approve patient 🡪 approve the patient’s registration to access the dashboard.
* Discharge patient 🡪 after appointment, admin can discharge the patient.

1. Appointment page 🡪 three boxes below will appear.

* View Appointment 🡪 admin can view all appointments.
* Book Appointment 🡪 admin can book appointments of registered patients.
* Approve Appointment 🡪 approve the patient’s appointment.

**DOCTOR:**

1. Patient page 🡪 two boxes below will appear.

* Your patient record 🡪 displays all of his patients’ record to perform the PTSD test with them.
* Your discharged patients 🡪 history of his discharged patients.

1. Appointment page 🡪 two boxes below will appear.

* Your appointment 🡪 view all his appointments with the patients.
* Delete appointment 🡪 can delete any appointment.

**PATIENT:**

1. Appointment page 🡪 can view his appointment with the doctor after admin has approved the appointment.
2. Discharge page 🡪 can view the discharge total bill after admin has generated the discharge bill and saved to the system.
3. Logout 🡪 admin, doctor and patient can log out from the system.

# Implementation



## Algorithm

Our project is ML based model integrated with a web-based platform. So the major algorithm that we used is given below (Banoula, 2023).

Table 64 Example of Algorithm

|  |
| --- |
| **Algorithm 1 Logistic Regression** |
| **Input:** X - matrix of size n x m (training examples x features)  y - vector of size n x 1 (target variable, 0 or 1)  learning\_rate - scalar (controls the step size in gradient descent)  num\_iterations - scalar (number of times to update weights using gradient descent) |
| **Output:** weights - vector of size m+1 x 1 (learned weights, including bias term) |
| 1: weights <- array of zeros of length m+1 // Initialize weights with zeros  2: X <- add\_bias\_term(X) // Add a bias term to the feature matrix X  3: sigmoid <- lambda z: 1 / (1 + exp(-z)) // Define the sigmoid function  4: **for** i in range(num\_iterations):  5: z <- dot(X, weights) // Calculate predicted y values using the current weights and feature matrix  6: y\_pred <- sigmoid(z)  7: error <- y\_pred – y // Calculate the error between predicted and true y values  8: gradient <- dot(X.T, error) / n // Update weights using gradient descent  9: weights <- weights - learning\_rate \* gradient  10: **return** weights |

## External APIs/SDKs

No third-party APIs/SDKs are used in the project implementation. Some examples of APIs in Django & ML used are provided here.

**Text Analytics:** If your Django ML-based website involves natural language processing, you may consider integrating text analytics APIs/SDKs such as Google Cloud Natural Language API, IBM Watson Natural Language Understanding, or Amazon Comprehend.

These are examples, but in this project, we have not implemented any third-party APIs/SDKs.

## User Interface

Details about user interface with descriptions. Provide the User Interface for each sub-system (such as Mobile App, Web App, Client App, Admin App). Provide description of each User Interface explaining the details.

When inserting User Interfaces, use appropriate size of the image, for example, for mobile app, 2-4 screens can be placed on a single page.

Following are few examples of User Interfaces:

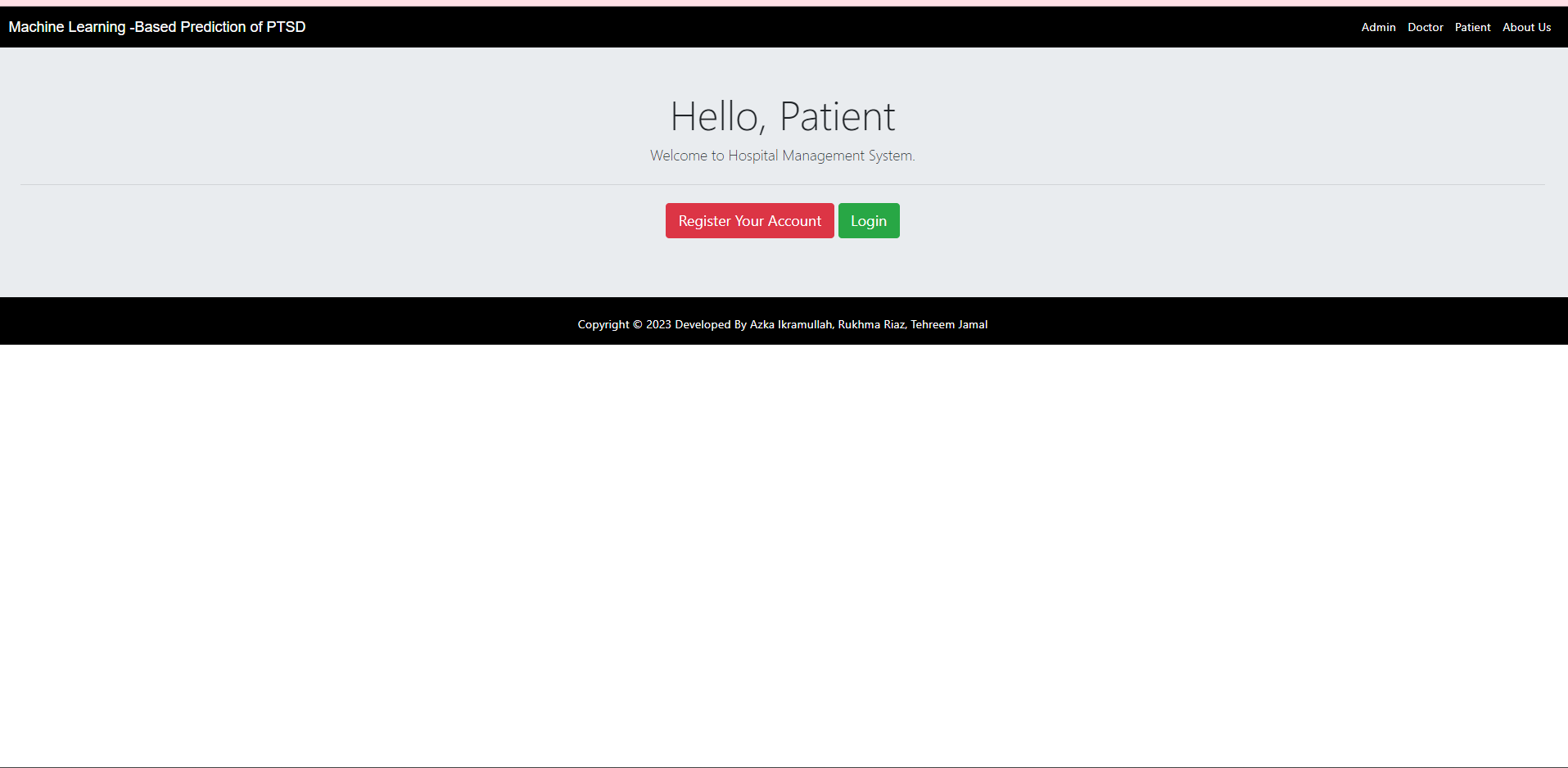
### Home Screen

Home screen of our website where user have to choose his/her role and act accordingly.

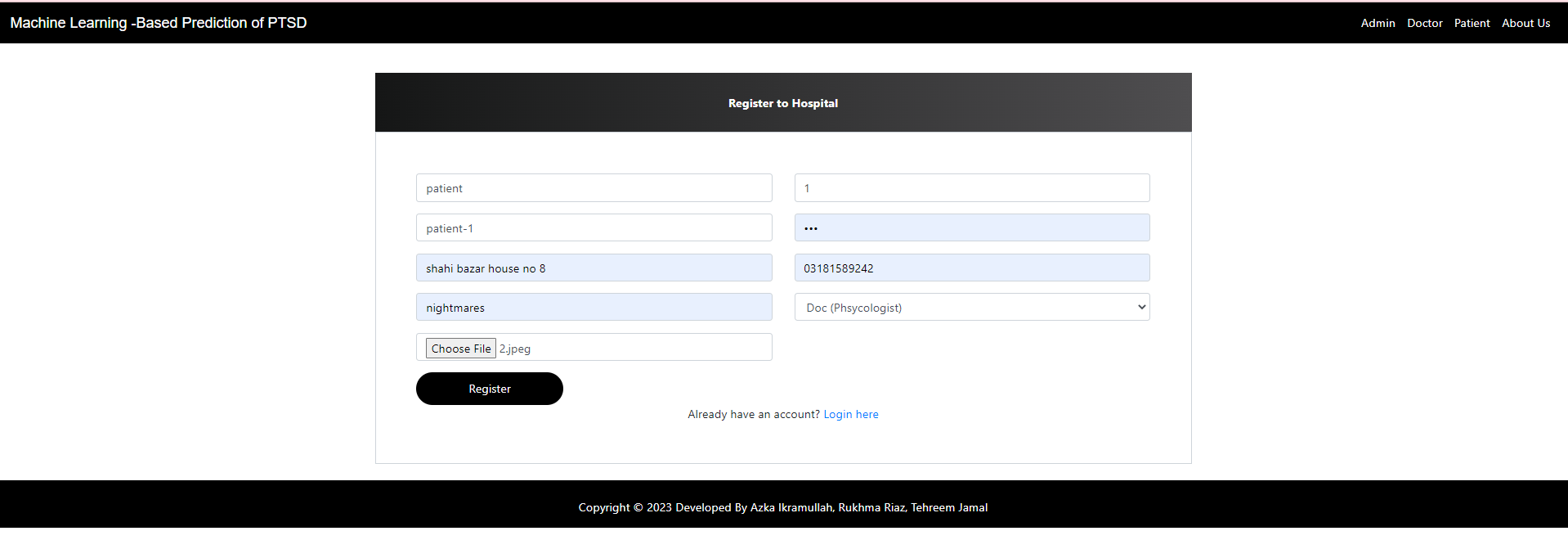
### 

### Patient

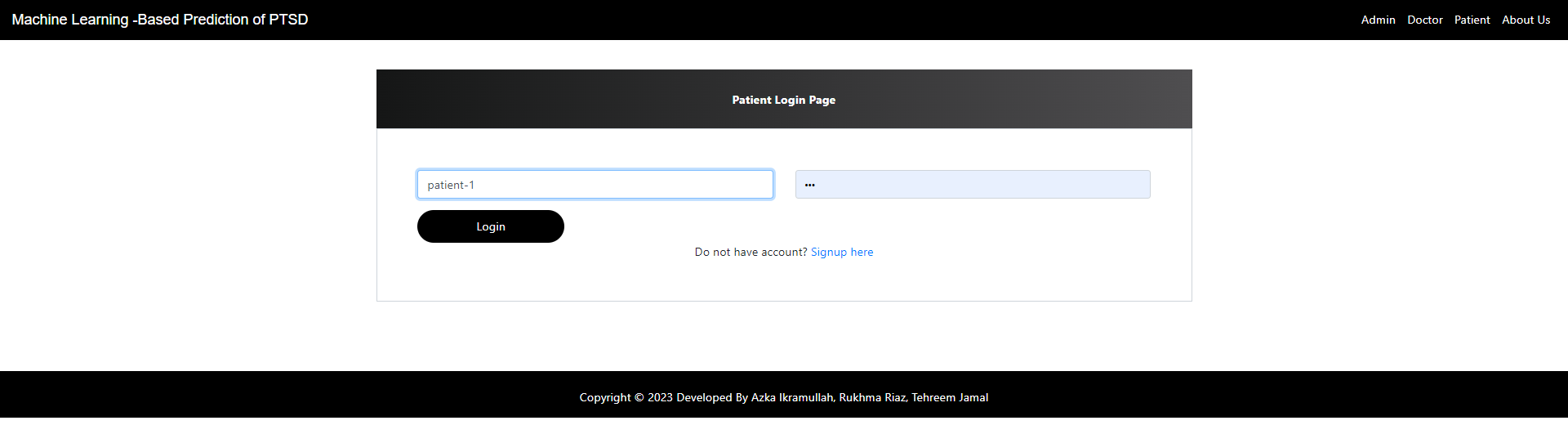
Patient visits the website, and click on book appointment button.



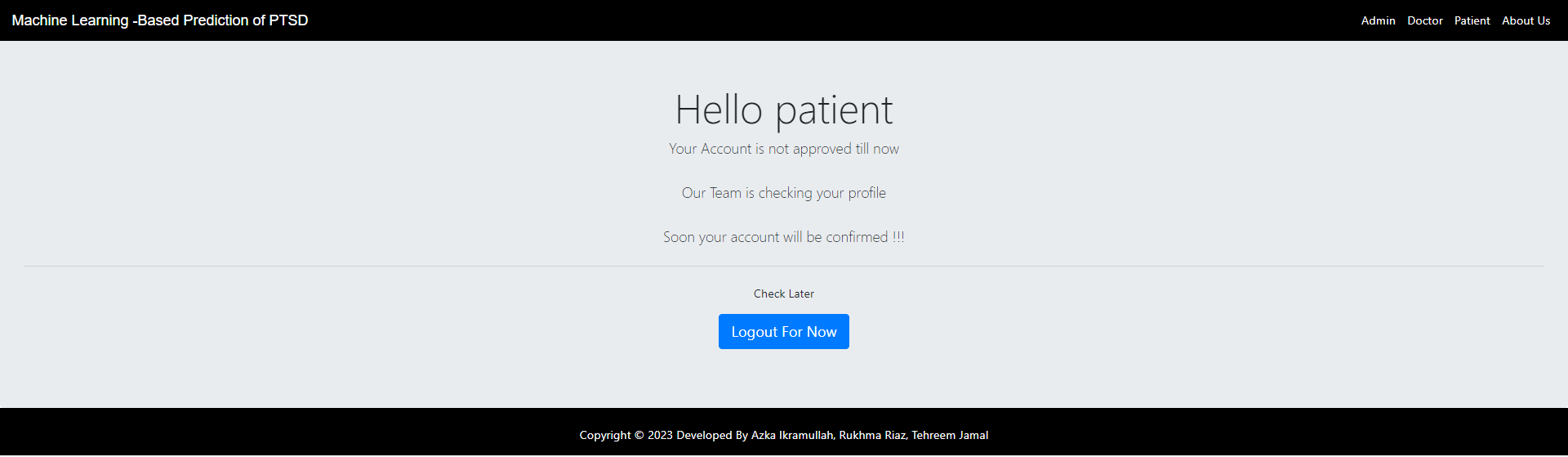
Patient registering his/her account for appointment



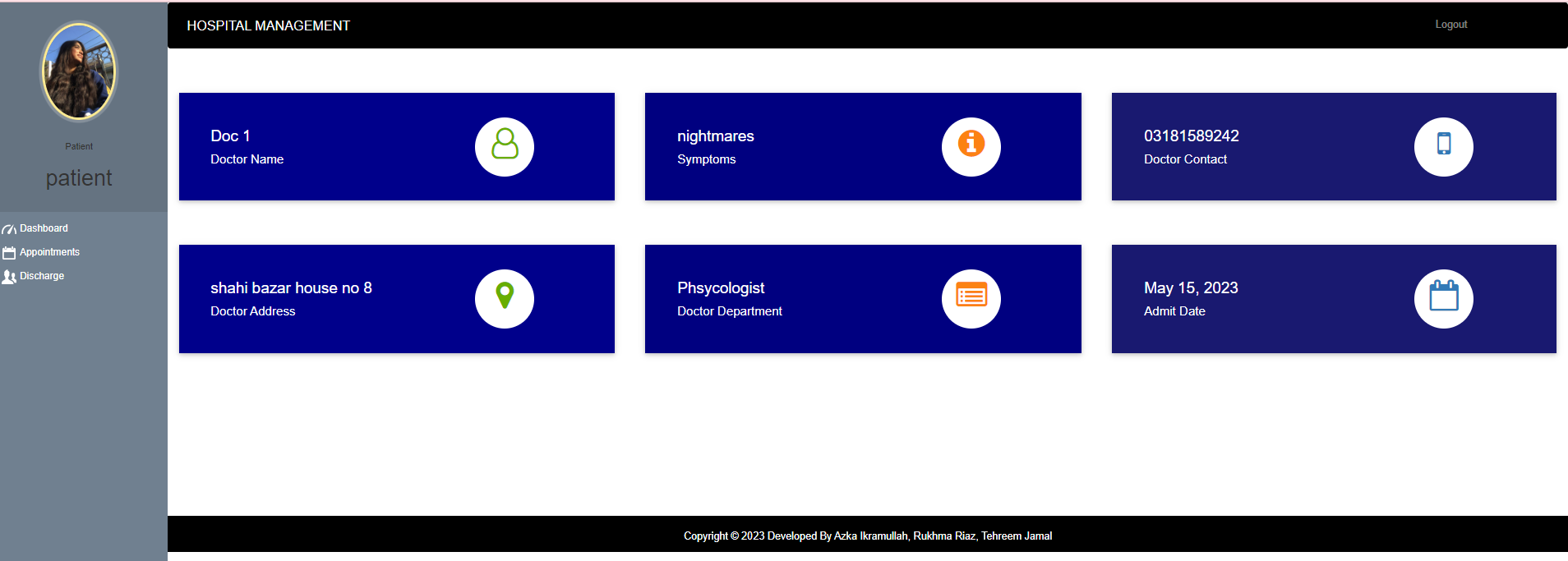
Patient tries to login,



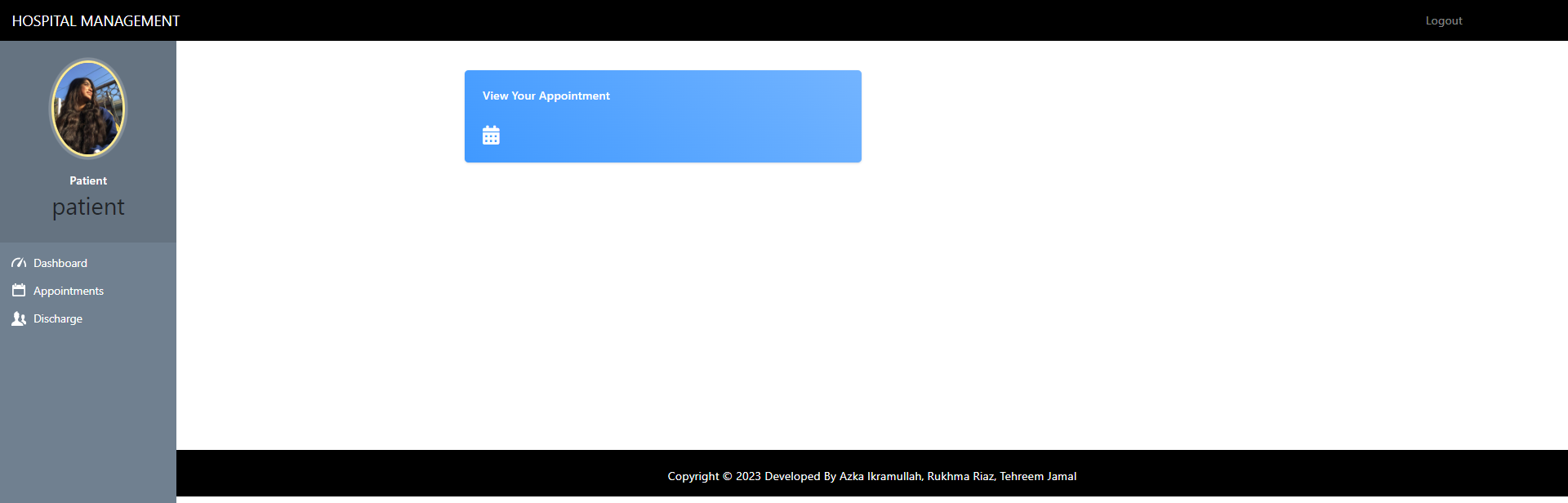
Patient tries to login, account is not yet approved by admin



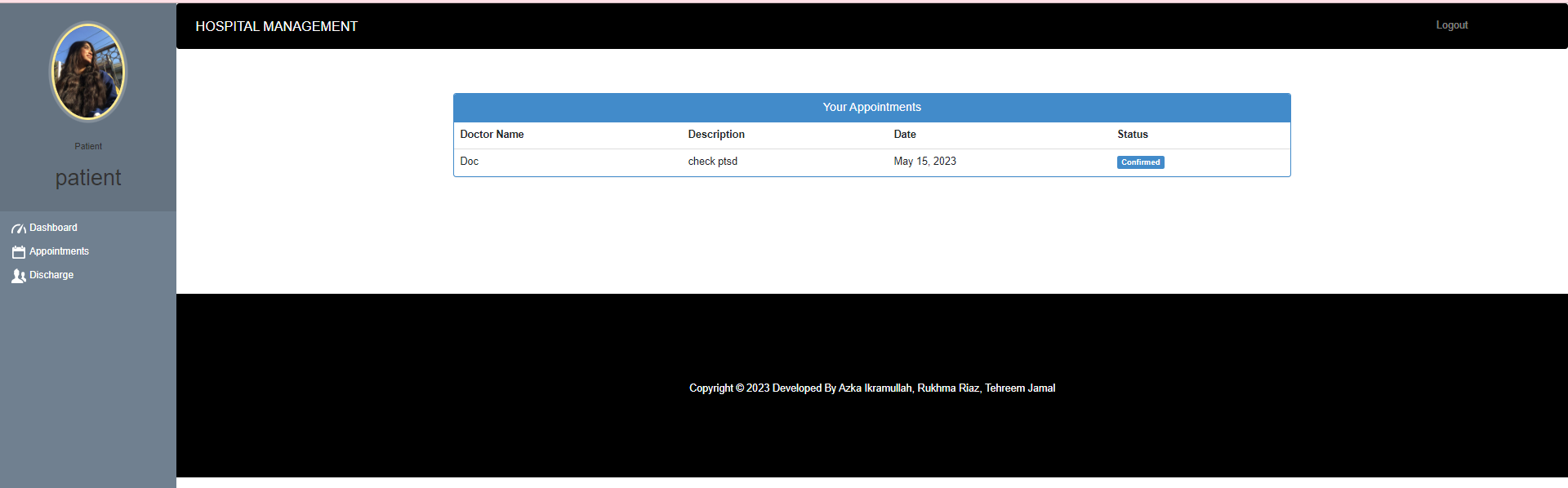
Dashboard of patient



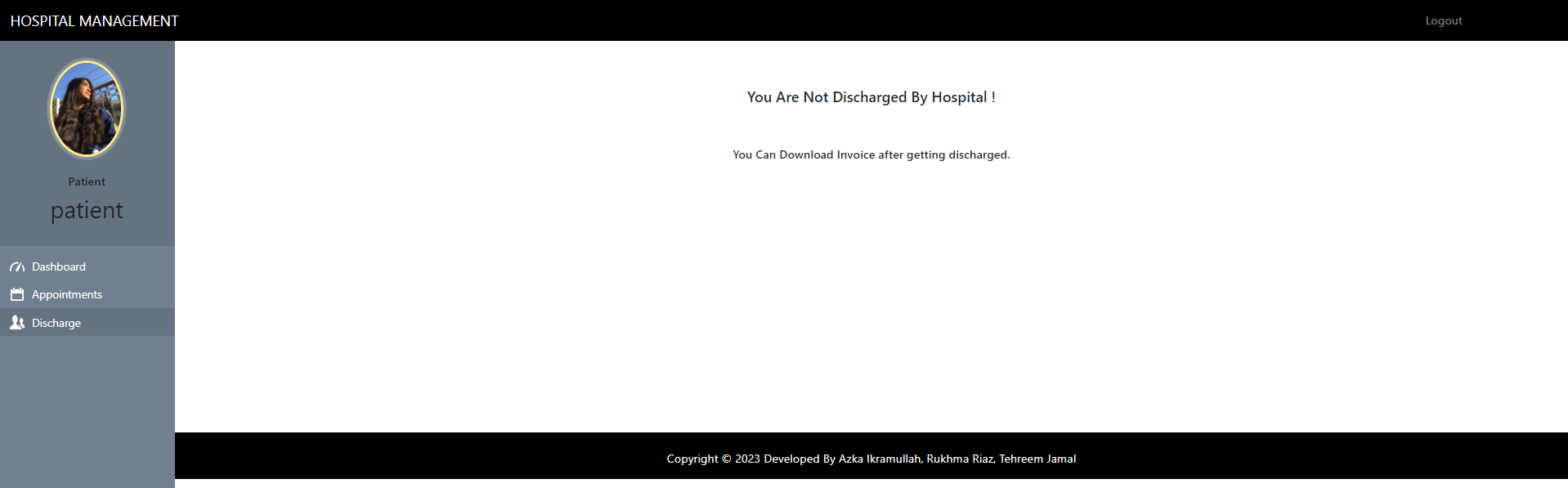
Patient can view his/her appointment



Patient can see his/her appointment is confirmed by admin

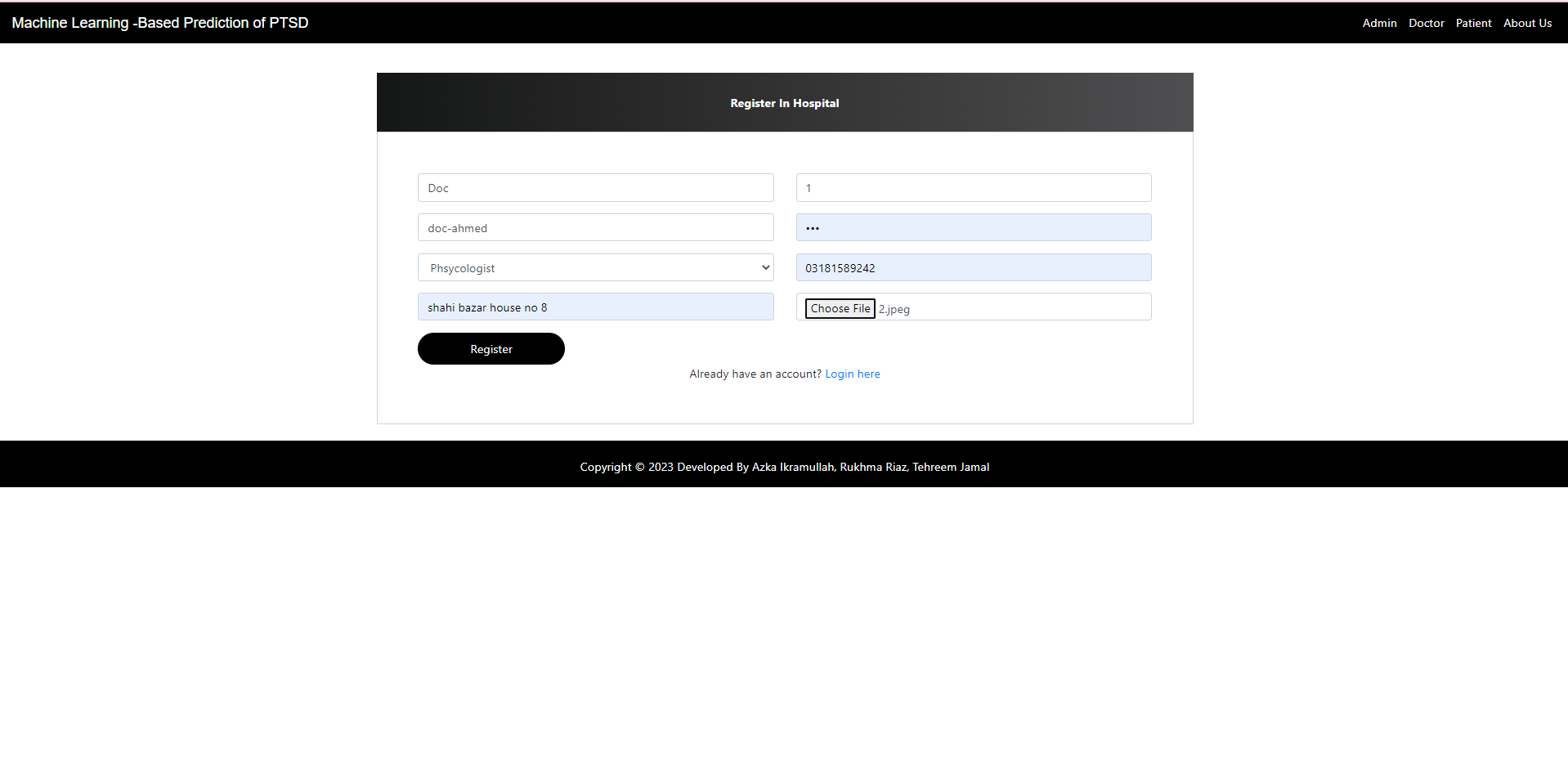


Patient can see the statistics on is he/she discharged by hospital or not

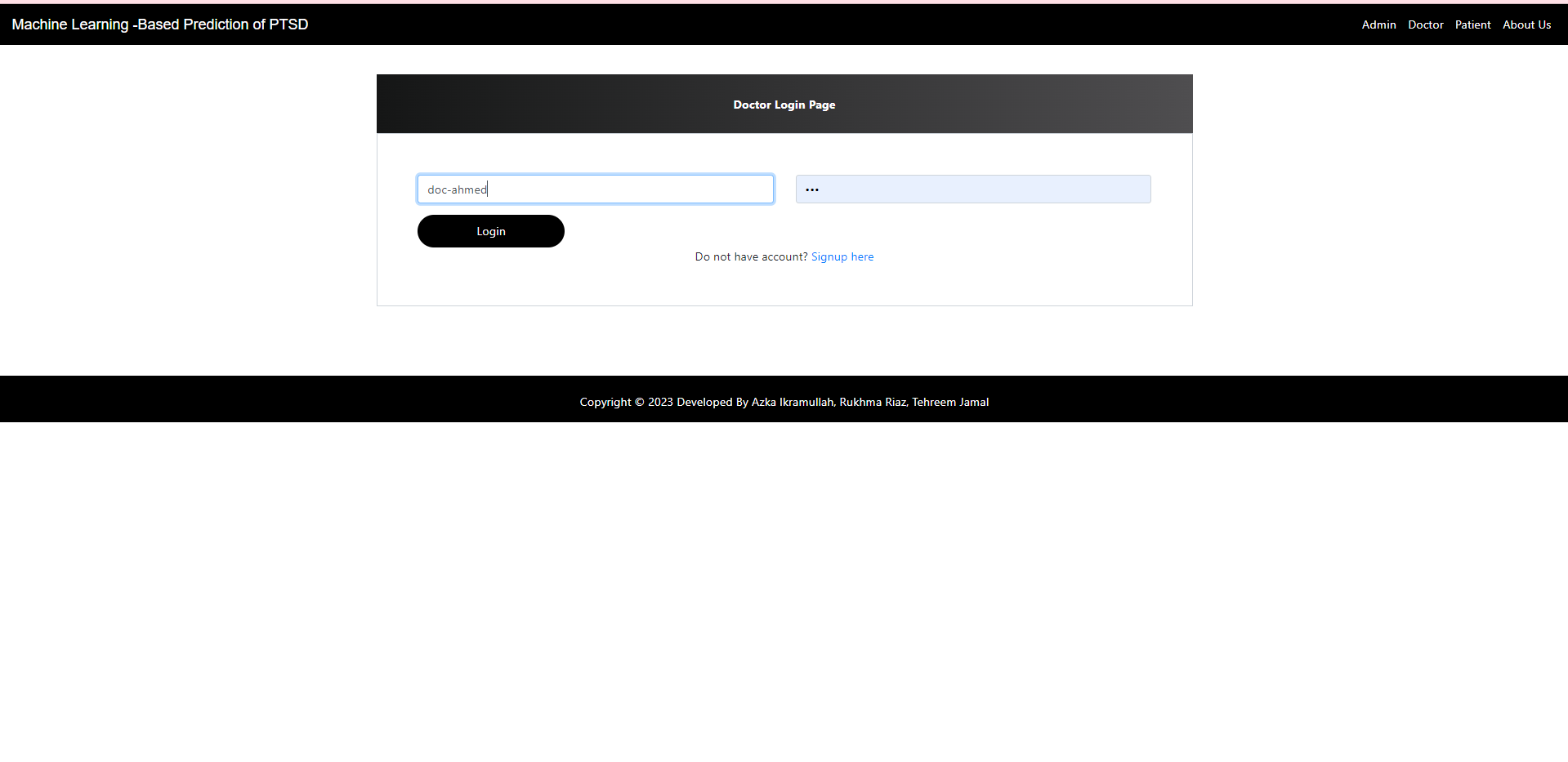


### Doctor

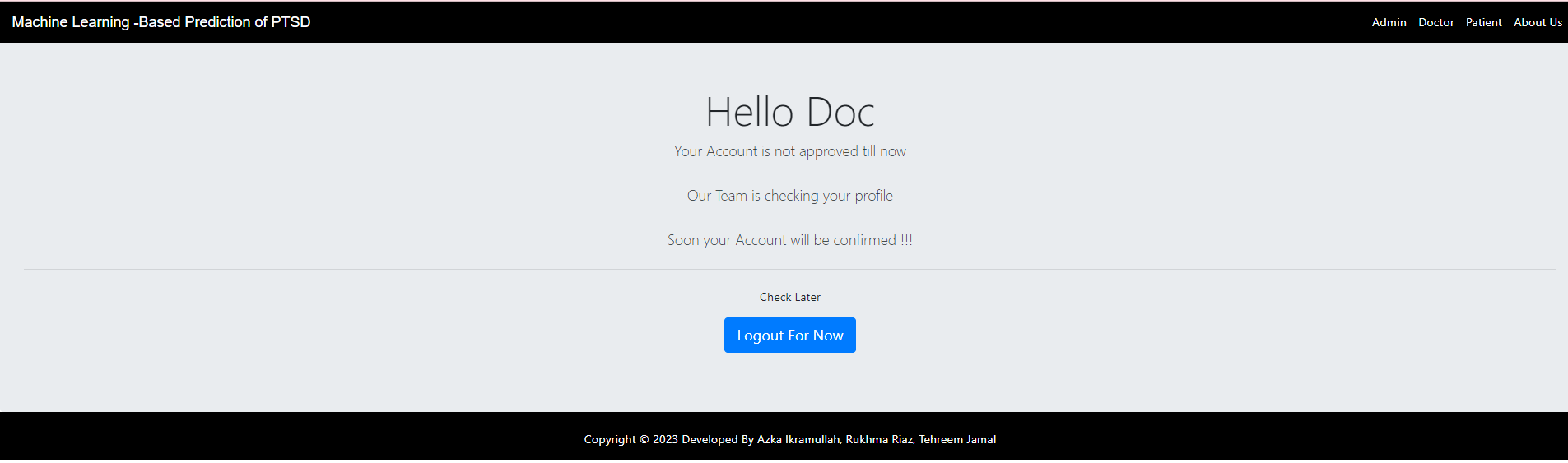
Doctor registering himself/herself for



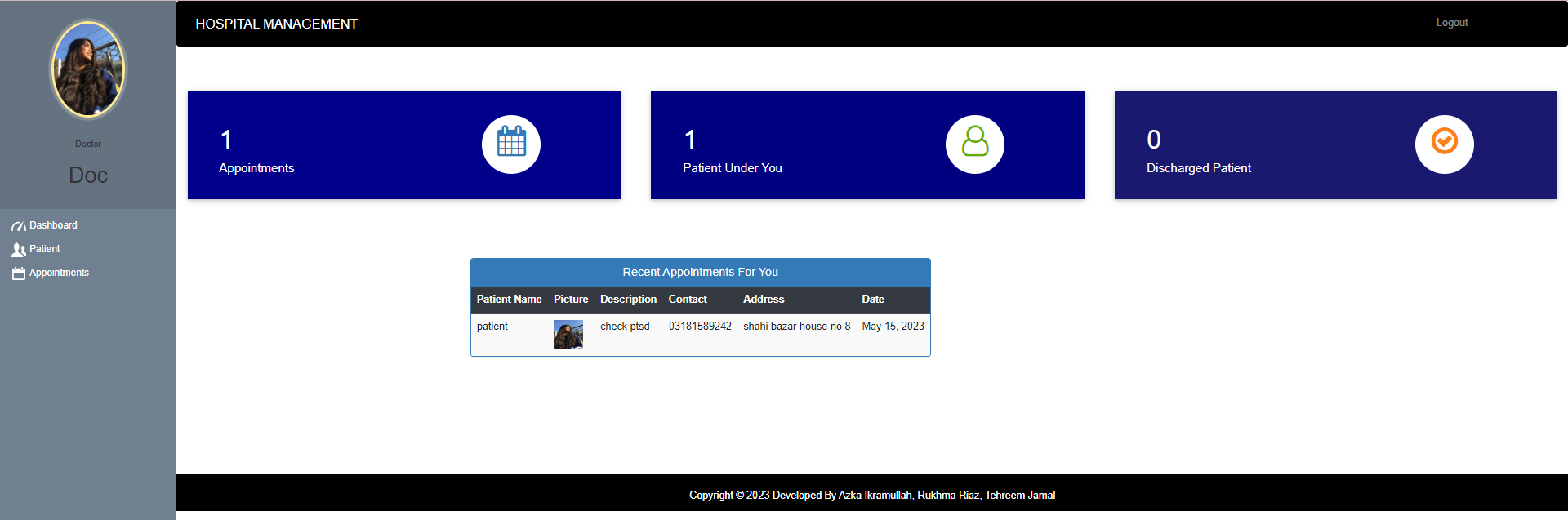
After entering his details, doctor go to login page



Doctor tried to login but his/her account is not approved from admin.



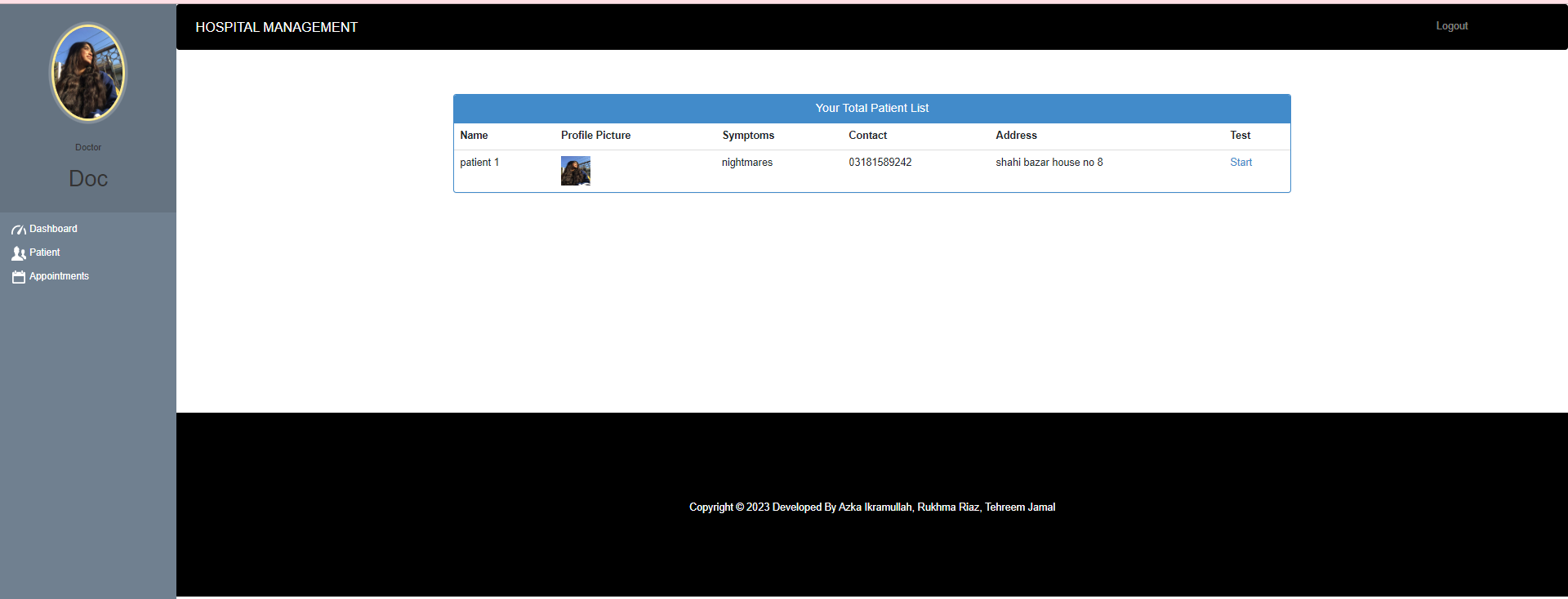
Doctor account is approved by admin, this is doctors dashboard. Doctor has one patient to check PTSD.



For patient doctor can perform these



Doctor can see patients under him/her



Doctor starts PTSD test of patient



Filling in the details of the questionnaire

A screenshot of a computer

Description automatically generated

If PTSD is detected

A picture containing text, software, multimedia software, computer icon

Description automatically generated

The doctor can advise the patient

A screenshot of a computer

Description automatically generated with medium confidence

After submitting advice, the doctor will save and download the prescription for the patient.

A white paper with black text

Description automatically generated with low confidence

In case if PTSD is not detected we get the following result.

A screenshot of a computer

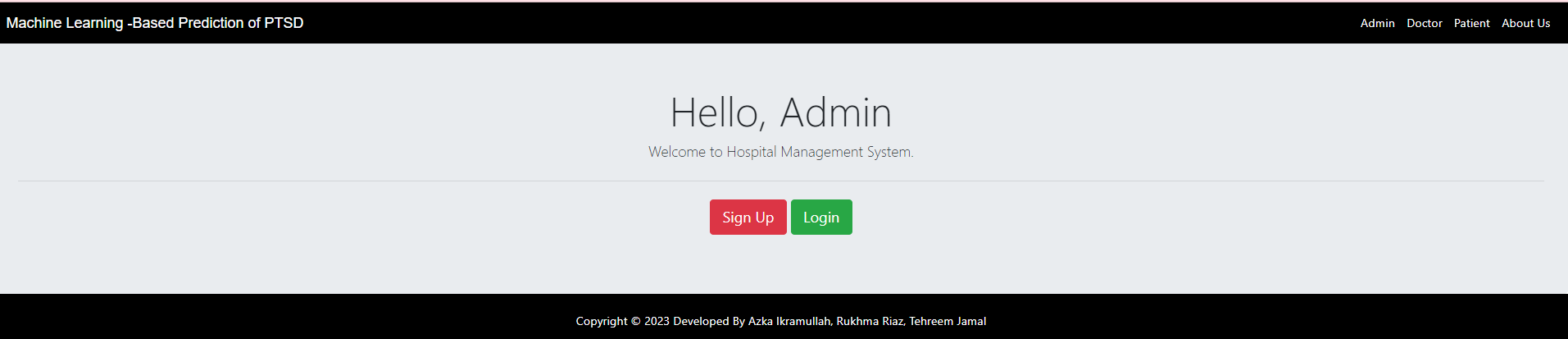
Description automatically generated with medium confidence

For appointment doctor can perform these

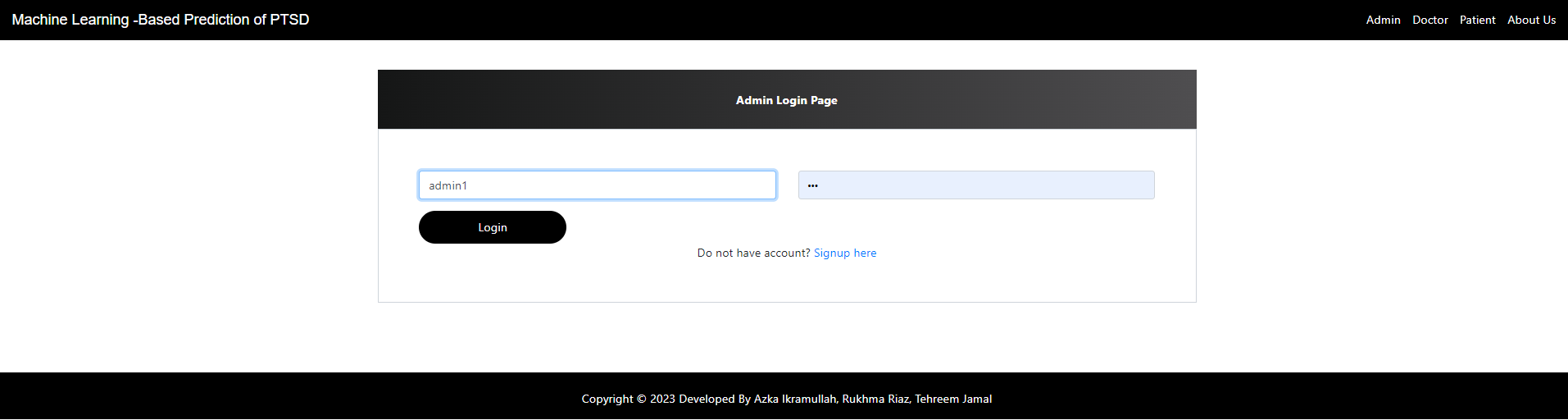


### Admin

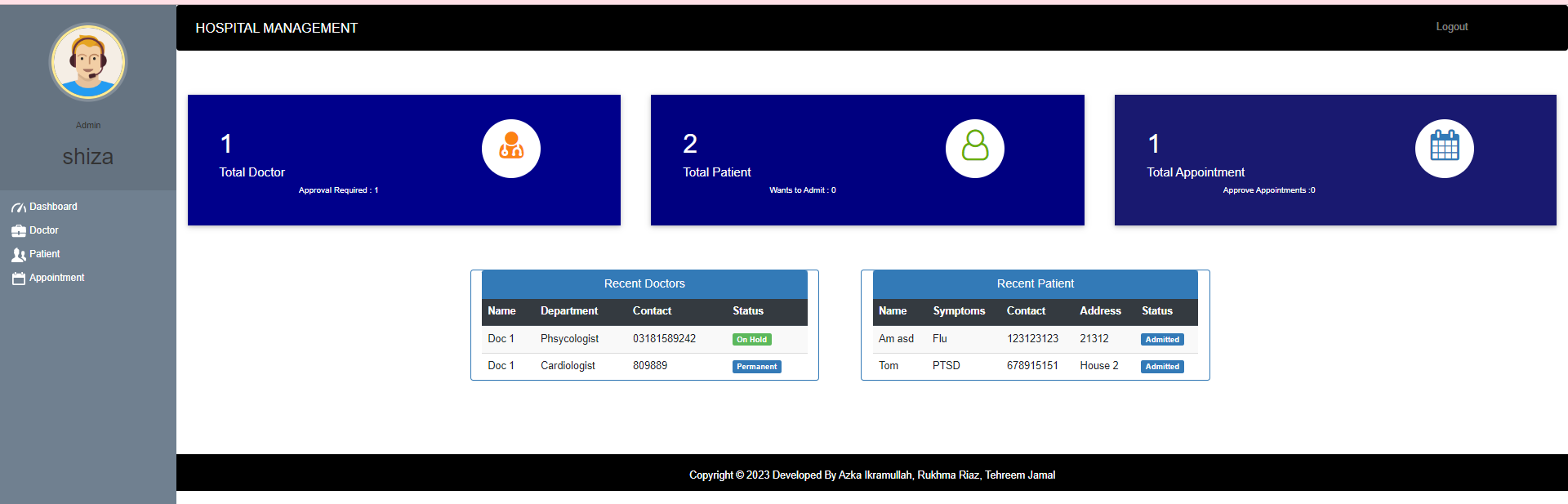
Admin can do login & signup.



Admin login to verify the doctor’s account.



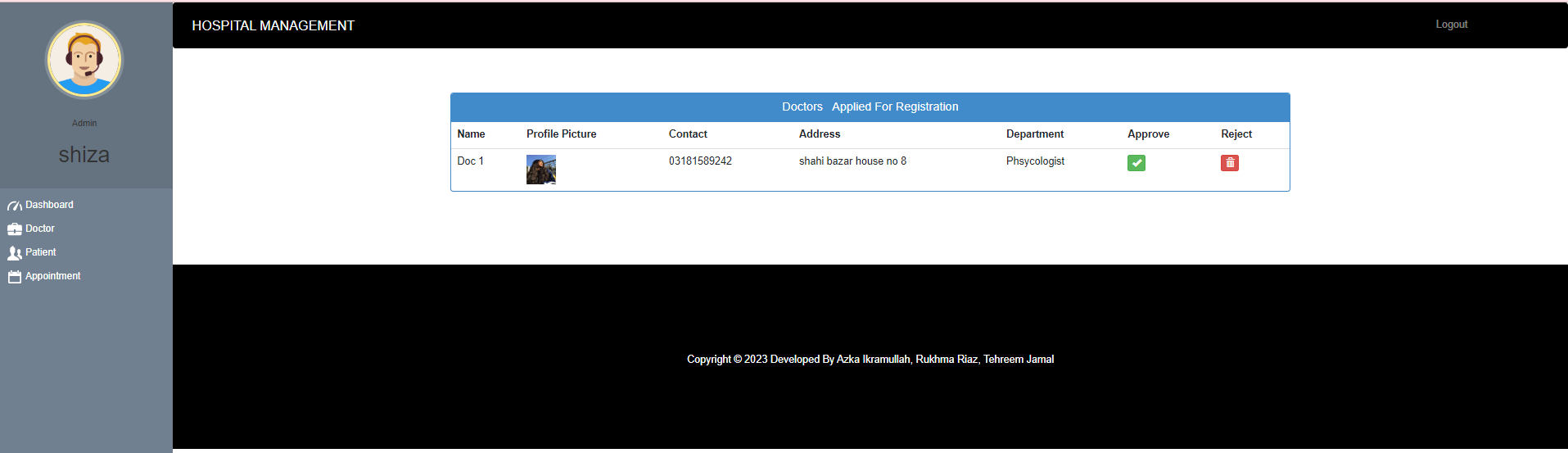
Admin dashboard



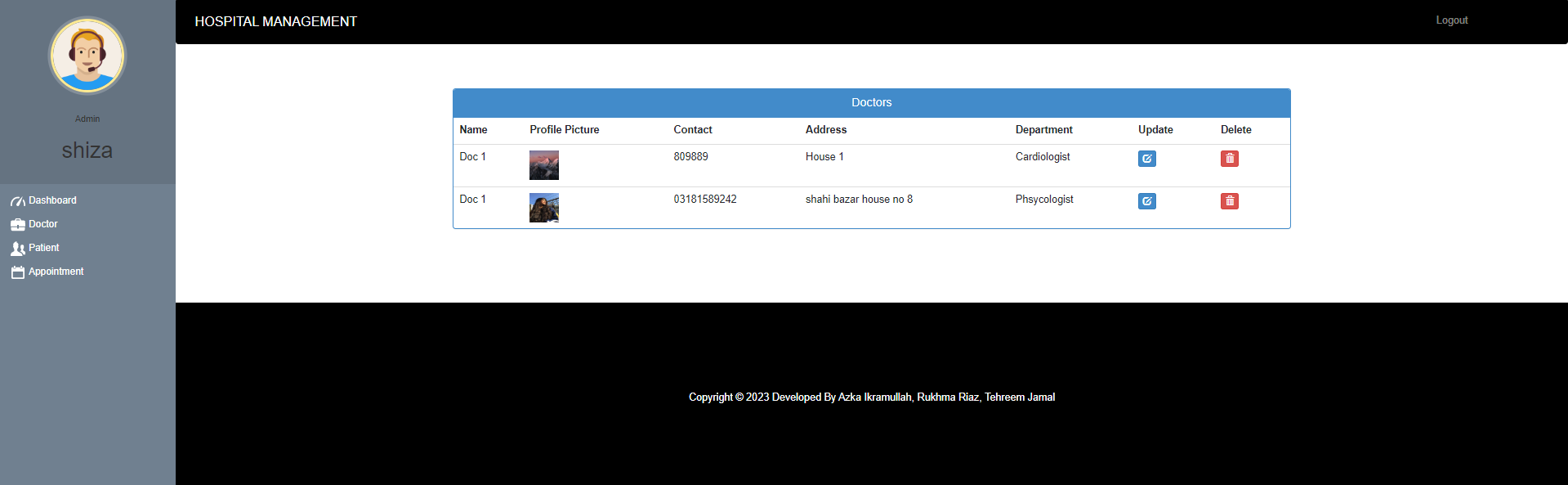
For doctor’s admin can perform these things



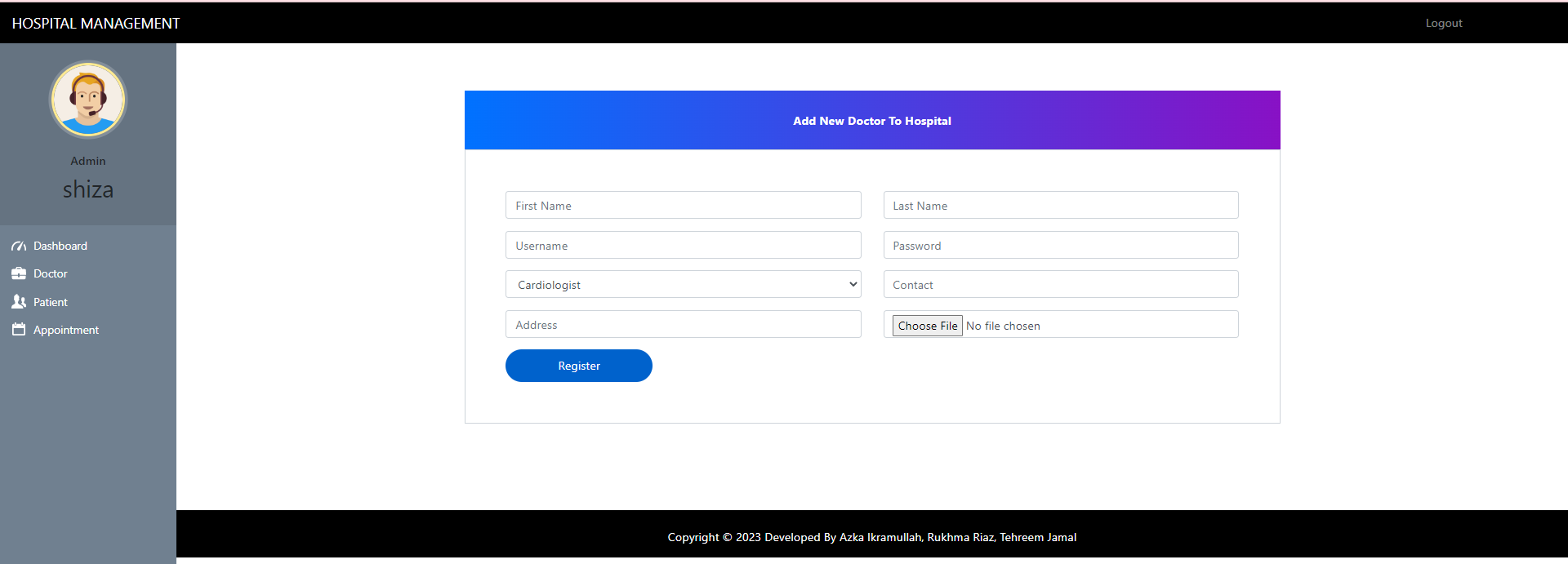
Admin approved the registered doctor



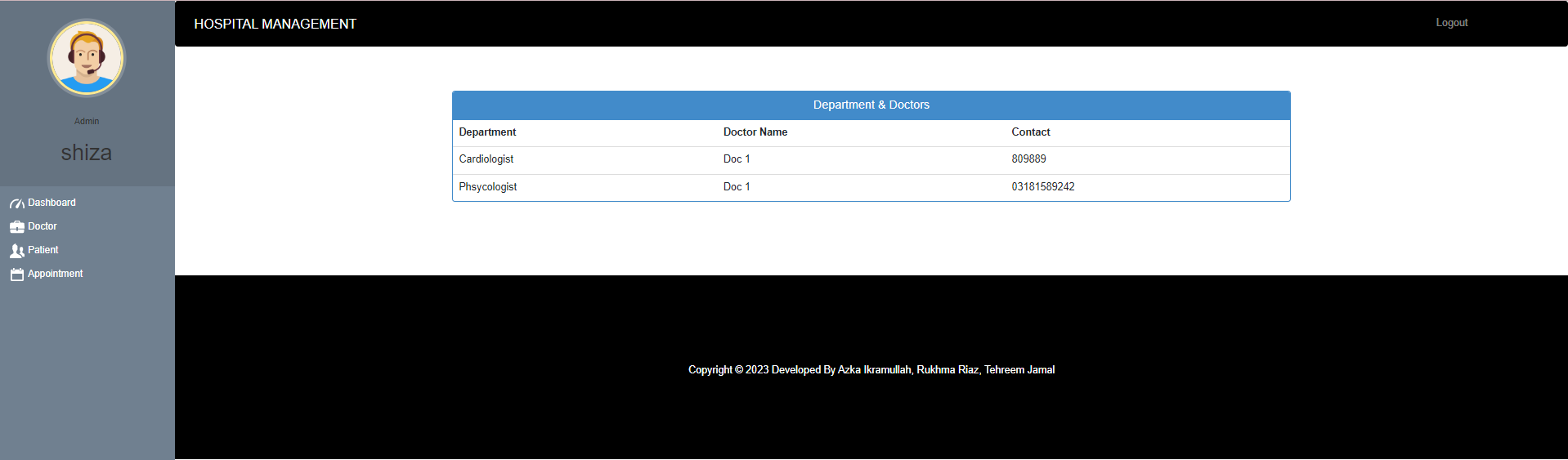
Now this doctor is added with doctor’s record



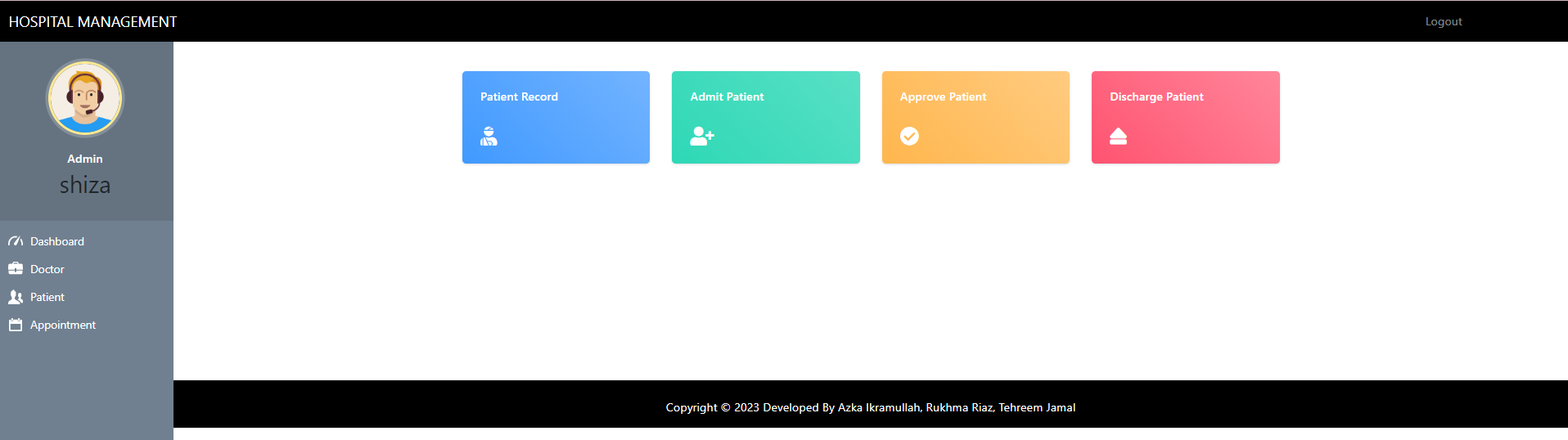
Admin can register doctor on his/her own too



Doctor’s with their specializations are shown like this



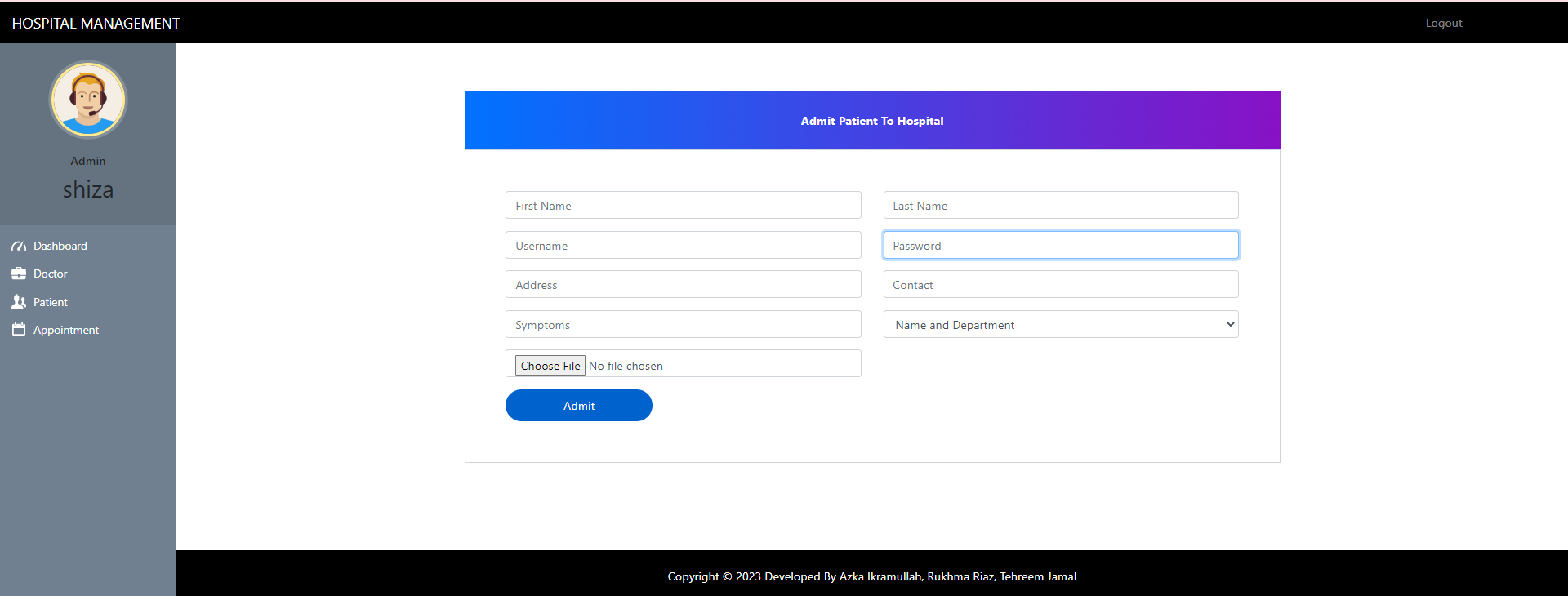
For patient’s admin can perform these



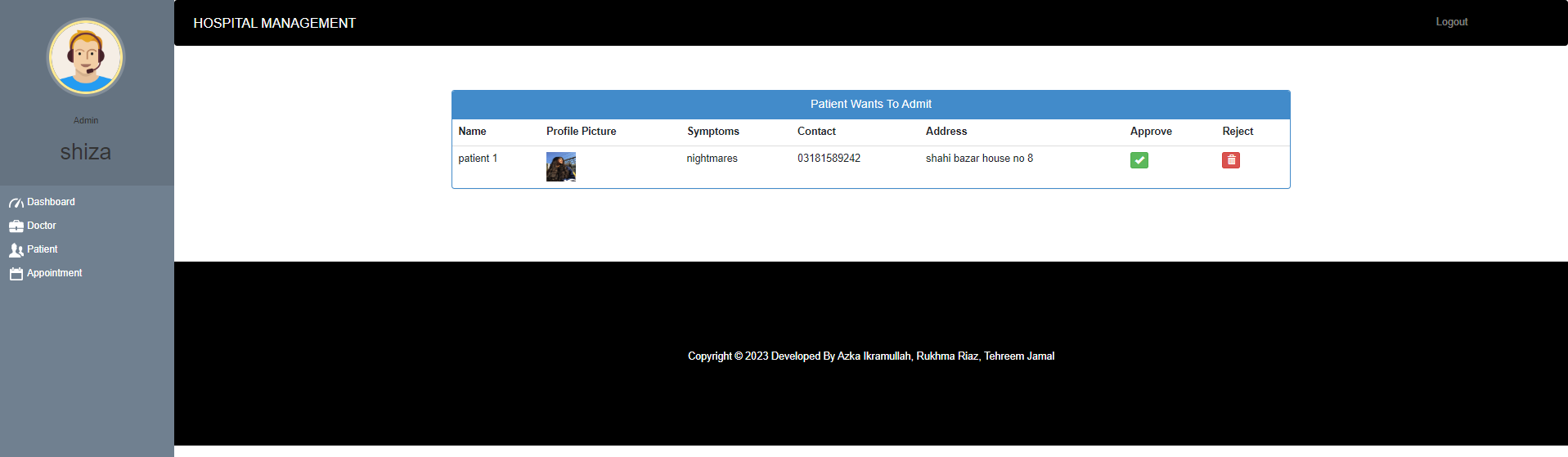
Total patients record can be seen here, admin can delete patient too



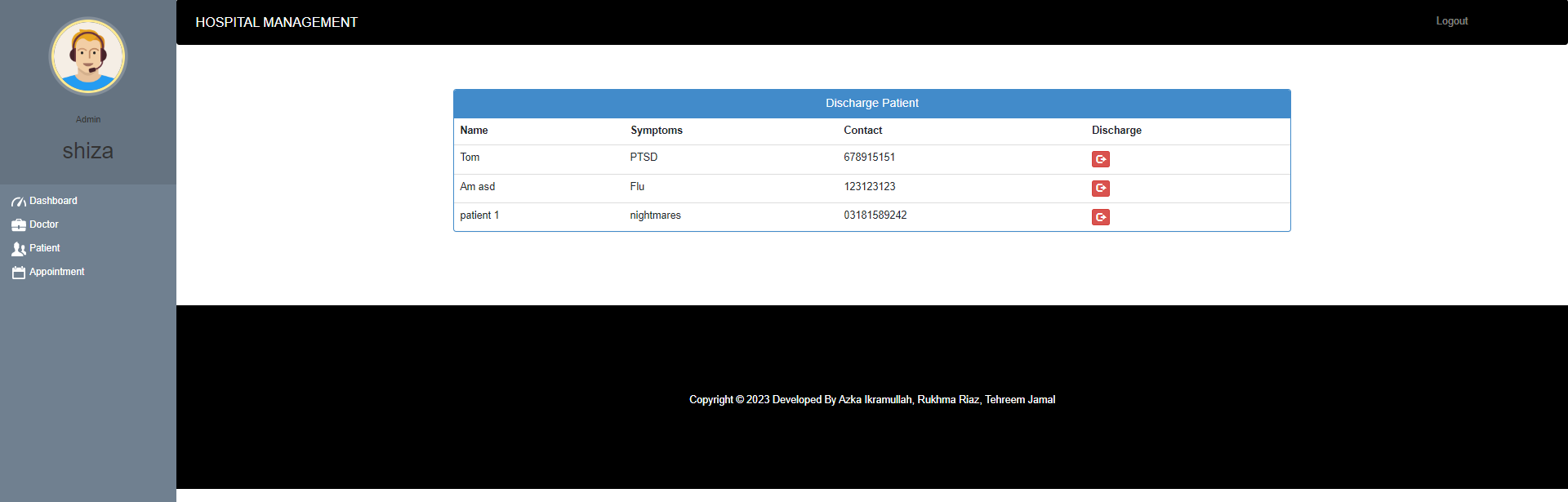
Admin can admit patient too



Admin approves the patient’s account



Admin can discharge patient



During discharge the bill is prompted

A screenshot of a computer

Description automatically generated with medium confidence

After putting in the details the total billing is calculated. The PDF can be downloaded as well. A screenshot of a computer

Description automatically generated with medium confidence

The downloaded bill is as below.

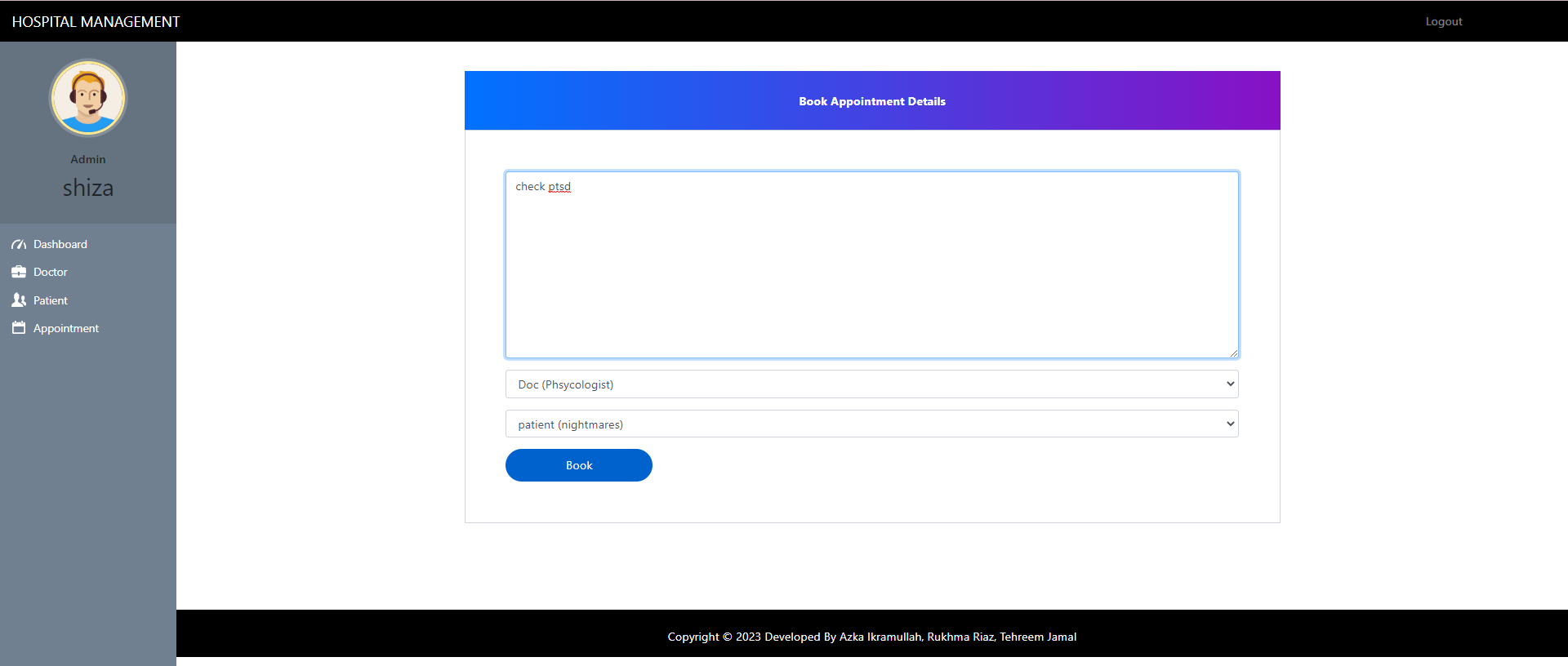
A screenshot of a medical form

Description automatically generated with low confidence

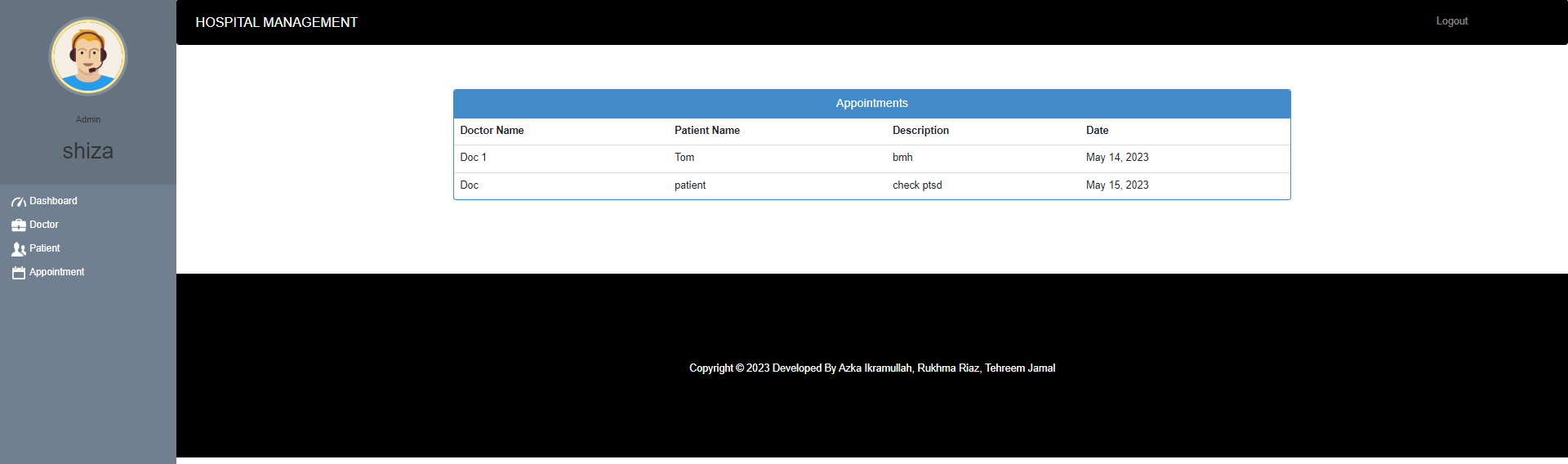
About appointments admin can do these things



Admin can book patient’s appointment to doctor



Admin can view total appointments



## Deployment

For deployment, we configured the local host. In development phase we normally deploy website on a local host. Once it is tested, verified and ready to use in market then we deploy it on any cloud or hosting platform. Also, when we will get some investors on this business idea as well as some customers then we will deploy it.

There are several hosting and cloud services available for deploying Django-based websites and machine learning models. Some popular options include:

* Heroku: Heroku is a cloud-based Platform-as-a-Service (PaaS) that supports Django-based websites and machine learning models. It offers an easy-to-use interface and supports several programming languages.
* Amazon Web Services (AWS): AWS is a cloud-based Infrastructure-as-a-Service (IaaS) that supports Django-based websites and machine learning models. It offers a wide range of services and features, including Elastic Beanstalk, which supports Python applications.
* Google Cloud Platform (GCP): GCP is a cloud-based Platform-as-a-Service (PaaS) that supports Django-based websites and machine learning models. It offers several services and features, including Google App Engine, which supports Python applications.
* Microsoft Azure: Azure is a cloud-based Platform-as-a-Service (PaaS) that supports Django-based websites and machine learning models. It offers several services and features, including Azure App Service, which supports Python applications.

Some important software and versions that are commonly used for Django-based websites and machine learning models include:

* Python: Python is the programming language used for developing Django-based websites and machine learning models. The current stable version of Python is Python 3.1.0.
* Django: Django is the web framework used for developing Django-based websites. The current stable version of Django is Django 3.2.4.
* Machine Learning Libraries: There are several machine learning libraries available for Python, including scikit-learn, TensorFlow, and PyTorch. The specific libraries and versions you use will depend on the requirements of your machine learning model.
* Web Server: There are several web servers available for hosting Django-based websites, including Apache and Nginx. The specific web server and version you use will depend on your deployment environment and requirements.

# Testing and Evaluation



Our system has been successfully developed, testing has to be performed to ensure that the system working as intended. This is also to check that the system meets the requirements stated earlier. Besides that, system testing will help in finding the errors that may be hidden from the user. The testing must be completed before it is deployed for use.

There are few types of testing which includes the unit testing, functional testing and integration testing (contributors, 2023).

## Unit Testing

It’s a level of software testing where individual units of a software/component are tested. The purpose is to validate that each unit of the software performs as designed.

**Unit Testing 1:** Login as Patient with valid and invalid credentials

**Testing Objective:** To ensure the login form is working correctly with valid and invalid credentials/inputs.

**Table 66 unit testing 1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Check the username field of login to validate that it takes proper username | Username: [patie](mailto:abc@gmail.com)nt | Validates username and move cursor to next textbox | Pass |
| 2 | Check the username field of login to validate that it displays error message. | Username: [patient2](mailto:abc@gmail.com) | Reloads the page to and prompt to enter again | Pass |

**Unit Testing 2:** Login as Doctor with valid and invalid credentials

**Testing Objective:** To ensure the login form is working correctly with valid and invalid credentials/inputs.

**Table 67 unit testing 2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Check the username field of login to validate that it takes proper username | Username: [doctor](mailto:abc@gmail.com) | Validates username and moves cursor to next textbox | Pass |
| 2 | Check the username field of login to validate that it displays error message. | Email: [patient2](mailto:abc@gmail.com) | Reloads the page to and prompt to enter again | Pass |

**Unit Testing 3:** Login as Admin with valid and invalid credentials

**Testing Objective:** To ensure the login form is working correctly with valid and invalid credentials/inputs.

**Table 68 unit testing 3**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Check the username field of login to validate that it takes proper username | Username: [azkaadmin](mailto:abc@gmail.com) | Validates username and moves cursor to next textbox | Pass |
| 2 | Check the username field of login to validate that it displays error message. | Email: [admin](mailto:abc@gmail.com) | Reloads the page to and prompt to enter again | Pass |

**Unit Testing 4:** Register as Patient with valid and invalid credentials.

**Testing Objective:** To ensure the signup form is working correctly with valid and invalid credentials/inputs.

**Table 69 unit testing 4**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Registering a new patient with the system to validate database entry | First Name:  [Tom](mailto:abc@gmail.com)  Last Name:  [Hard](mailto:abc@gmail.com)  Username:  [patient1](mailto:abc@gmail.com)  Password:  [patient](mailto:abc@gmail.com)  Address:  [House](mailto:abc@gmail.com) # 3, Street 4  Contact Number:  [9023424](mailto:abc@gmail.com)  Symptoms:  [Headache](mailto:abc@gmail.com)  Name and Department:  [Doc](mailto:abc@gmail.com) & Psychologist  Profile Picture:  [Picture.jpeg](mailto:abc@gmail.com) | Validates all the required credentials and stores them in the database. | Pass |
| 2 | Registering a new patient with the system to validate database entry if any credential is missed | First Name:  [Tom](mailto:abc@gmail.com)  Last Name:  [Hard](mailto:abc@gmail.com)  Username:  [patient1](mailto:abc@gmail.com)  Password:  [patient](mailto:abc@gmail.com)  Address:  [House](mailto:abc@gmail.com) # 3, Street 4  Contact Number:  [9023424](mailto:abc@gmail.com)  Symptoms:  [Headache](mailto:abc@gmail.com)  Name and Department:  (Empty)  Profile Picture:  [Picture.jpeg](mailto:abc@gmail.com) | Highlights the field and prompts to enter the required value. | Pass |

**Unit Testing 5:** Register as Doctor with valid and invalid credentials.

**Testing Objective:** To ensure the signup form is working correctly with valid and invalid credentials/inputs.

**Table 70 unit testing 5**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Registering a new doctor with the system to validate database entry | First Name:  [Doc](mailto:abc@gmail.com)  Last Name:  [One](mailto:abc@gmail.com)  Username:  [doctor](mailto:abc@gmail.com)  Password:  [doctor](mailto:abc@gmail.com)  Department:  Psychologist  Contact Number:  [9023424](mailto:abc@gmail.com)  Address:  [House](mailto:abc@gmail.com) # 3, Street 4  Profile Picture:  [Picture.jpeg](mailto:abc@gmail.com) | Validates all the required credentials and stores them in the database. | Pass |
| 2 | Registering a new doctor with the system to validate database entry if any credential is missed | First Name:  [Doc](mailto:abc@gmail.com)  Last Name:  [One](mailto:abc@gmail.com)  Username:  [doctor](mailto:abc@gmail.com)  Password:  [doctor](mailto:abc@gmail.com)  Department:  Psychologist  Contact Number:  (Empty)  Address:  [House](mailto:abc@gmail.com) # 3, Street 4  Profile Picture:  [Picture.jpeg](mailto:abc@gmail.com) | Highlights the field and prompts to enter the required value. | Pass |

**Unit Testing 6:** Register as Admin with valid and invalid credentials.

**Testing Objective:** To ensure the signup form is working correctly with valid and invalid credentials/inputs.

**Table 71 unit testing 6**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Registering a new admin with the system to validate database entry | First Name:  [admin](mailto:abc@gmail.com)  Last Name:  [Jhon](mailto:abc@gmail.com)  Username:  [Steven](mailto:abc@gmail.com)  Password:  [admin1](mailto:abc@gmail.com) | Validates all the required credentials and stores them in the database. | Pass |
| 2 | Registering a new admin with the system to validate database entry if any credential is missed | First Name:  [admin](mailto:abc@gmail.com)  Last Name:  (Empty)  Username:  [Steven](mailto:abc@gmail.com)  Password:  [admin1](mailto:abc@gmail.com) | Highlights the field and prompts to enter the required value. | Pass |

**Unit Testing 7:** Booking appointment as Admin with valid and invalid credentials.

**Testing Objective:** To ensure the booking form is working correctly with valid and invalid credentials/inputs.

**Table 72 unit testing 7**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Booking a new appointment with the system to validate database entry | Description:  [Having](mailto:abc@gmail.com) sever headaches  Doctor Name and Department:  [Doctor](mailto:abc@gmail.com)1 (Psychologist)  Patient Name and Symptoms:  [Tom](mailto:abc@gmail.com) (PTSD) | Validates all the required credentials and stores them in the database. | Pass |
| 2 | Booking a new appointment with the system to validate database entry if any credential is missed | Description:  (Empty)  Doctor Name and Department:  [Doctor](mailto:abc@gmail.com)1 (Psychologist)  Patient Name and Symptoms:  [Tom](mailto:abc@gmail.com) (PTSD) | Highlights the field and prompts to enter the required value. | Pass |

**Unit Testing 8:** Filling the PTSD Questionnaire with valid and invalid credentials.

**Testing Objective:** To ensure the questionnaire form is working correctly with valid and invalid credentials/inputs.

**Table 73 unit testing 8**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Completing the PTSD questionnaire with the details to get prediction | Gender:  [Male](mailto:abc@gmail.com)  Age:  [23](mailto:abc@gmail.com)  Value of ISS:  [2](mailto:abc@gmail.com)  Suffering from Inj\_Vio:  [Yes](mailto:abc@gmail.com)  Headache:  [Yes](mailto:abc@gmail.com)  Symptoms of PCA:  [No](mailto:abc@gmail.com)  Morphine:  [10](mailto:abc@gmail.com)  Benzo: [No](mailto:abc@gmail.com)  Days Passed:  [14](mailto:abc@gmail.com)  Level of Ketamine:  [10](mailto:abc@gmail.com)  Pulse Rate:  [95](mailto:abc@gmail.com) | Validates all the required credentials and stores them in the csv. | Pass |
| 2 | Completing the PTSD questionnaire with the details to get prediction | Gender:  [Male](mailto:abc@gmail.com)  Age:  [(Empty)](mailto:abc@gmail.com)  Value of ISS:  [2](mailto:abc@gmail.com)  Suffering from Inj\_Vio:  [Yes](mailto:abc@gmail.com)  Headache:  [Yes](mailto:abc@gmail.com)  Symptoms of PCA:  [No](mailto:abc@gmail.com)  Morphine:  [10](mailto:abc@gmail.com)  Benzo: [No](mailto:abc@gmail.com)  Days Passed:  [14](mailto:abc@gmail.com)  Level of Ketamine:  [10](mailto:abc@gmail.com)  Pulse Rate:  [95](mailto:abc@gmail.com) | Validates all the required credentials and stores them in the csv. | Pass |

## Functional Testing

In this functional testing, the functionality of each of the module is tested. This is to ensure that the system produced meets the specifications and requirements.

**Functional Testing 1:** Login with different roles (Admin, Patient, Doctor)

**Objective**: To ensure that the correct page with the correct navigation bar is loaded.

**Table 74 functional testing 1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** | **Result** |
| 1. | Login as an ‘Admin member. | Username: (correct username azka)  Password:  (correct password azkaadmin) | Main page for the Admin is loaded with the Admin navigation bar. | Logged in and redirected to admin main page. | Pass |
| 2. | Login as a ‘Doctor’ member. | Username: doctor  Password:  doctor | Main page for the Doctor is loaded with the doctor navigation bar. | Logged in and redirected to admin main page. | Pass |
| 3. | Login as a ‘Patient’ member. | Username: patient  Password:  patient | Main page for the Patient is loaded with the doctor navigation bar. | Logged in and redirected to patient main page. | Pass |
| 4. | Login as a ‘Doctor’ member. | Username: doctor  Password:  patient | Main page for the Doctor is loaded with the doctor navigation bar. | Login failed – invalid credentials error | Fail |

**Functional Testing 2:** Registering different roles (Admin, Patient, Doctor)

**Objective**: To ensure that the registered account is available in respected panels.

**Table 75 functional testing 2**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** | **Result** |
| 1. | Register as an ‘Admin member. | Username: (correct username azka)  Password:  (correct password azkaadmin) | Main login page is reloaded with the account being shown on the admin dashboard. | Page reloaded and once logged in accounts are displayed on the main dashboard. | Pass |
| 2. | Register as a ‘Doctor’ member. | Username: doctor  Password:  doctor | Main login page is reloaded with the account being shown on the doctor dashboard. | Page reloaded and once logged in accounts are displayed on the main dashboard. | Pass |
| 3. | Register as a ‘Patient’ member. | Username: patient  Password:  patient | Main login page is reloaded with the account being shown on the patient dashboard. | Page reloaded and once logged in accounts are displayed on the main dashboard. | Pass |

**Functional Testing 3:** Testing all the patient management features.

**Objective**: To ensure that actions related to managing patients are working fine.

**Table 76 functional testing 3**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** | **Result** |
| 1. | Scheduling appointment with a doctor | Description:  [Having](mailto:abc@gmail.com) sever headaches  Doctor Name and Department:  [Doctor](mailto:abc@gmail.com)1 (Psychologist)  Patient Name and Symptoms:  [Tom](mailto:abc@gmail.com) (PTSD) | Appointment for the respective patient with the corresponding doctor is booked. | Appointment is booked and is reflected in the doctor’s appointment schedule along with patient’s schedule. | Pass |
| 2. | Admitting a patient into the hospital. | First Name:  [Tom](mailto:abc@gmail.com)  Last Name:  [Hard](mailto:abc@gmail.com)  Username:  [patient1](mailto:abc@gmail.com)  Password:  [patient](mailto:abc@gmail.com)  Address:  [House](mailto:abc@gmail.com) # 3, Street 4  Contact Number:  [9023424](mailto:abc@gmail.com)  Symptoms:  [Headache](mailto:abc@gmail.com)  Name and Department:  [Doc](mailto:abc@gmail.com) & Psychologist  Profile Picture:  [Picture.jpeg](mailto:abc@gmail.com) | Patient is admitted into the system and is reflected across the databases. | Page reloaded and once logged in accounts are displayed on the main dashboard. | Pass |
| 3. | Discharge a patient. | Discharge button in front of the patient is pressed. | Details of the patient are removed from record and page is reloaded. | Page reloaded and details are removed. | Pass |

**Functional Testing 4:** Testing all the doctor related features.

**Objective**: To ensure that actions related to doctors’ actions are working fine.

**Table 77 functional testing 4**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** | **Result** |
| 1. | Taking the PTSD Test of the patient. | Gender:  [Male](mailto:abc@gmail.com)  Age:  [23](mailto:abc@gmail.com)  Value of ISS:  [2](mailto:abc@gmail.com)  Suffering from Inj\_Vio:  [Yes](mailto:abc@gmail.com)  Headache:  [Yes](mailto:abc@gmail.com)  Symptoms of PCA:  [No](mailto:abc@gmail.com)  Morphine:  [10](mailto:abc@gmail.com)  Benzo: [No](mailto:abc@gmail.com)  Days Passed:  [14](mailto:abc@gmail.com)  Level of Ketamine:  [10](mailto:abc@gmail.com)  Pulse Rate:  [95](mailto:abc@gmail.com) | Based on the input provided the system returns the result. | We are prompted whether the patient has PTSD or not. | Pass |
| 2. | Deleting appointment | The red cancel button in front of a booked appointment is pressed. | Appointment is cancelled. | The patient’s appointment is cancelled and related changes are made in the database. | Pass |

**Functional Testing 4:** Generating invoice for the patient after being discharged.

**Objective**: To ensure that financial calculations are correct, and invoice is generated.

**Table 78 functional testing 4**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** | **Result** |
| 1. | Generating invoice of a discharged patient | Using the red button to discharge the patient. | The invoice is generated after entering some final details of the patient’s stay. | Invoice is ready to download as PDF and to view. | Pass |

## Business Rules Testing

Methodology for creating decision table:

Table

Description automatically generated

**Rule 1: New users need to create an account.**

**Rule 2: Old users can login**

**Rule 3: Admin needs to approve account.**

**Rule 4: Approved accounts can access dashboards.**

**Rule 5: Have an appointment.**

**Rule 6: Admitted into hospital**

**Rule 7: Admitted patients can give test.**

**Rule 8: Receive prescription**

**Policy 1**

* New users need to create an account
* Old user can login

**Table 79 policy 1**

|  |  |  |
| --- | --- | --- |
| Condition | Rule 1 | Rule 2 |
| New User | Y | N |
| Old User | Y | N |

**Policy 2**

* New users need to create an account.
* Old user can login
* Approved accounts can access dashboards.

**Table 80 policy 2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Condition | Rule 1 | Rule 2 | Rule 3 | Rule 4 |
| New User | Y | N | N | N |
| Can login? | N | Y | - | N |
| Approved Account | N | Y | Y | Y |

**Policy 3**

* New users need to create an account.
* Old user can login
* Approved accounts can access dashboards and book appointments.
* Admitted patient can take PTSD Test.

**Table 81 policy 3**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Condition | Rule 1 | Rule 2 | Rule 3 | Rule 4 | Rule 5 | Rule 6 | Rule 7 | Rule 8 |
| New User | Y | N | N | N | N | N | N | N |
| Can login? | N | Y | Y | N | N | N | N | N |
| Approved Account | N | Y | Y | Y | Y | N | N | N |
| Book appointment | N | Y | Y | Y | Y | Y | Y | N |
| Get Admit | N | Y | Y | Y | Y | Y | N | N |
| Take PTSD Test | N | Y | Y | Y | Y | Y | Y | Y |
| Receive Prescription | N | N | N | Y | Y | Y | Y | Y |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Condition | Rule 1 | Rule 2 | Rule 3 | Rule 4 | Rule 5 | Rule 6 |
| New User | Y | N | N | N | N | N |
| Can login? | N | Y | N | N | N | N |
| Approved Account | N | Y | Y | N | N | N |
| Book appointment | N | Y | Y | Y | Y | N |
| Get Admit | N | Y | Y | Y | N | N |
| Take PTSD Test | N | Y | Y | Y | Y | Y |
| Receive Prescription | N | N | Y | Y | Y | Y |

Rules 2 and 3 have the same values so we can combine them, and written collectively in Rule 2 column. Also Rule 4 and Rule 5 can be combined as well, written collectively in Rule 4 column.

## Integration Testing

**Integration Testing 1:** Scheduling Patient Appointment

**Testing Objective:** To ensure the scheduling is being done correctly and *the* ***interface*** *between* module ‘Patient/Doctor Management’ and module ‘Appointment/Scheduling’ *is running correctly*.

**Table 82 integration testing 1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** | **Result** |
| 1. | Make Appointment | Doctor schedule, patient preferred date and time | Successfully create doctor-patient appointment showing date and time of appointment. | Appointment created successfully | Pass |
| 2. | Delete appointment. | Select the appointment | Appointment slot will be deleted. | Appointment deleted successfully | Pass |

**Integration Testing 2:** Advising patient after getting result.

**Testing Objective:** To ensure the after getting a result we can advise the patient accordingly.

**Table 83 integration testing 2**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result.** | **Actual result** | **Result** |
| 1. | Give advice to patient | After being detected with PTSD the doctor can give advice to patient. | Successfully create doctor-patient appointment showing date and time of appointment. | Appointment created successfully | Pass |
| 2. | Downloading Advice report | Select download option on advice page. | PDF is downloaded. | The report is downloaded as a PDF. | Pass |

# Conclusion and Future Work



## Conclusion

In this project we implemented a hospital management system that catered to all users such as patient, administrator, and doctor. The system was able to handle all requests that were received. From managing accounts of different users, booking appointments, deleting appointments, using key elements to define whether a patient was suffering from PTSD and then the doctor was able to advise the patient accordingly. Our PTSD prediction system used linear regression to predict using the patient’s data which was gathered using the questionnaire form at the time of the test (Likarenko, 2020). If the patient was diagnosed with PTSD, the doctor was able to give advice and recommend tests for it. The advice report as well as the financial reports were generated in PDF format for the user to save in the local drive or print for the patient. Our system served as an automated solution for hospitals to detect PTSD using Machine learning based algorithm. The model used for the prediction had a high accuracy of 98.14% on testing data. Combining all these modules makes our system a very credible and robust software solution.

## Future Work

The current PTSD prediction model is based on a set number of questions which are asked from the user, in future we can work on increasing the questions which will help us better understand and catch PTSD traces in patient. We can also move towards stepping away from the questionnaire approach and incorporate more advanced feature extraction techniques such as heart rate monitoring, brain waves analyzing and many other biological features. These features will not only make our system more accurate but will also allow patients to simply take the tests instead of the doctor asking them about their symptoms (Bourla, Mouchabac, Hage, & Ferreri, 2018).

# References

Banoula, M. (2023, February 22). *An Introduction to Logistic Regression in Python*. Retrieved from SimpleiLearn: https://www.simplilearn.com/tutorials/machine-learning-tutorial/logistic-regression-in-python

Bourla, A., Mouchabac, S., Hage, W. E., & Ferreri, F. (2018). e-PTSD: an overview on how new technologies can improve prediction and assessment of Posttraumatic Stress Disorder (PTSD). *European journal of psychotraumatology*.

contributors, M. (2023, February 24). *Django Tutorial Part 10: Testing a Django web application*. Retrieved from Mdn Web Docs: https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Testing

Foundation, D. S. (2023, May 12). *Django Documentation.* Retrieved from Django: https://docs.djangoproject.com/en/4.2/intro/tutorial02/

Group, C. C., Simon, N., Robertson, L., Lewis, C., Roberts, N. P., Bethell, A., . . . Bisson, J. I. (2021). Internet‐based cognitive and behavioural therapies for post‐traumatic stress disorder (PTSD) in adults. *The Cochrane database of systematic reviews*.

Kolk, B. v. (2000). Posttraumatic stress disorder and the nature of trauma. *Dialogues in clinical neuroscience*, 7–22.

Likarenko, Y. (2020). *Hospital Management Software: All-In-One Guide.* Up Tech.

Mueller-Pfeiffer, C., Schick, M., Schulte-Vels, T., O'Gorman, R., & Michels, L. (2013). Atypical visual processing in posttraumatic stress disorder. *NeuroImage. Clinical*, 531–538.

Schultebraucks, K., Yadav, V., Shalev, A. Y., Bonanno, G. A., & Galatzer-Levy, I. R. (2022). Deep learning-based classification of posttraumatic stress disorder and depression following trauma utilizing visual and auditory markers of arousal and mood. *Psychological medicine*, 957–967.

Taylor-Desir, M. (2022). *What is Posttraumatic Stress Disorder (PTSD)?* Washington: American Psychiatric Association.

Wshah, S., Skalka, C., & Price, M. (2019). Predicting Posttraumatic Stress Disorder Risk: A Machine Learning Approach. *MIR mental health*, e13946.