**Acknowledgement**

We would like to express our special thanks of gratitude to (Md Shahed Talukder) as our project manager who gave us the golden opportunity to do this wonderful project on the topic **“Digital Sheba”,** which also helped us doing a lot of research and we come to know about so many new things. We really thankful to him.

Secondly, we would also like to thank our teammates (Johirul Islam Joy, Md Jony Ahmed and Sultana Akter Surovi) who helped us a lot in finishing this project within the limited which helped us increase our knowledge and skills. We are making this project not only for marks but to also increase our knowledge and for further implementation as a real-world solution of a problem.

Thanks again to all who helped us.

**Abstract**

**“Digital Sheba”** is a web based & mobile based application that is focused on the patient who admitted to the hospital. It will be able to schedule the time of taking medicine and getting that equipment according to the patients need. Our system includes registration of patients, storing their details into the system and also computerized billing sections. Finally, this app focuses on giving the best service of caring from the nurses and alarming it to the patient who is admitted to the hospital. Our system includes registration of patients, storing their details into the system and also computerized billing sections.

**Table of Contents**

**CONTENTS** --------------------------------------------------------------------------- Page

Board of examiners I – II

Declaration III

Acknowledgements IV

Abstract V

**CHAPTER**

**CHAPTER 1: Introduction 5-7**

1.1 Introduction 5

1.2 Motivation 6

1.3 Objectives 6

1.4 Expected Outcomes 7

**CHAPTER 2: BACKGROUND 8-10**

2.1 Introduction 8

2.2 Related Works 8-9

2.3 Comparative Studies 9

2.4 Scope and Features 10

2.5 Challenges 10

**CHAPTER 3: Requirement Specification 11-34**

3.1 Requirement Collection & Analysis 11-12

3.2 Use Case Modeling & Design 12-32

3.3 Logical Model Data 33

3.4 Design Requirements 34

**CHAPTER 4: Design Specification 35-50**

4.1 Front-End Design 35-36

4.2 Back-End Design 36-37

4.3 Interaction Design & UX 38

4.4 Implementation 38-39

**CHAPTER 5: Implementation & Testing 39-70**

5.1 Implementation & Database 39

5.2 Implementation of Front-End & Design 39-54

5.3 Implementation of Interaction 55

5.4 Testing Implementation 55

5.5 Test Result & Report 56

**CHAPTER 6: Conclusion & Future Scope 57-58**

6.1 Discussion & Conclusion 57

6.2 Scope for Further Developments 58

**LIST OF FIGURES**

**Figures Page No**

Figure 3.3.1 Use Case Diagram – Admin 13

Figure 3.3.2 Use Case Diagram – Doctor 14

Figure 3.3.3 Use Case Diagram – Nurse 15

Figure 3.3.4 Use Case Diagram – Assistant 16

Figure 3.3.5 Use Case Diagram – Patient 17

Figure 3.3.6 Schema Diagram | Logical Data Model 33

Figure 3.3.7 Django Design Pattern 37

Figure 3.3.8 Entity Relationship Diagram | Implementation of Database 39

Figure 3.3.9 User Interface of the System 40-54

**CHAPTER 1: INTRODUCTION**

* 1. **Introduction**

Many people are affected by many types of diseases every day, for which they admit to a hospital. But as we know, the circumstances of our hospitals are so dire nowadays. We fall into many problems regarding being admitted into a hospital, taking seats for the patient, or taking good hospital members' good services. We fall into trials not only before accepting a patient also after admitting them too. Sometimes it takes so much time to call a nurse to look after the patient. Also, we fall into problems during taking equipment that we need for our patients. If we analysis so deeply we can see that not only patients but also doctors and nurses fall into so many problems, like making the schedule for a patient about what they need, maintaining time of giving medicine, assigning nurses to take care of them, visiting hour of nurses and so on. In short, the system of a hospital not appropriately organized for which both hospital members and patients fall into so many problems; and as a result, hospital members are failing to give full effort to patients, and patients are not getting their service correctly, and this situation is going out of our hand day by day. As we know, an effectively performing health sector is an essential precondition for society's overall development. But there have approximately 60,000 doctors and a deficit of almost 140,000 nurses. Here, the worst part is that there is only one nurse for every three physicians. If those hospitals' system is not well organized, then it's impossible to give their best to those patients, and finally, it will make chaos.

As we know Hospitals are the essential part of our lives, providing best medical facilities to people suffering from various ailments, which may be due to change in climatic conditions, increased work-load, emotional trauma stress etc. It is necessary for the hospitals to keep track of its day-to-day activities & records of its patients, doctors, nurses, assistants and other staff personals that keep the hospital running smoothly & successfully.

But Keeping track of all the activities and their record on paper is very cumbersome and error prone. It also is very inefficient and a time-consuming process Observing the continuous increase in population and number of people visiting the hospital. Recording and maintaining all these records is highly unreliable, inefficient and error-prone. It is also not economically & technically feasible to maintain these records on paper. Thus, keeping the working of the manual system as the basis of our project. We have developed an automated version of the manual system, names as **“Digital Sheba”.**

* 1. **Motivation**

The problem faced by the patients in Bangladesh is making them unbearable for a hospital's whole organizing system. They are facing so many issues to get the proper service they need. Patients face problems getting good service, and hospital members like doctors, nurses, and assistance face the problem while serving their patients properly because of the system's lack. "Digital Sheba" is a platform that has come for giving help to patients and members of the hospitals. It will help doctors to provide their patients' proper equipment correctly at the correct time when it needs. It will help nurses to see their schedule at any time. As we know, this platform's primary focus is to serve patients properly; they will be able to check that they are correctly filled or not. This platform will give them the ability to provide feedback about the assigned nurse. Finally, in short, the platform will help both users give and get the best of it.

* 1. **Objectives**

The objective of the project targets:

* To help the doctor to maintain the schedule of giving the proper treatment to the patient.
* To help nurses to keep the time of their own which serving the patient.
* To create an appropriate program for giving medicine and equipment.
* To help patients of getting adequate treatment from hospital members.
  1. **Expected Outcomes**

The expected outcome from the successful implementation of the project:

* Patients will be able to see their medical equipment list and the schedule of medicine.
* Doctors will be able to assign nurses for the patients and also will be able to set which medicine will need for those patients.
* Nurses will be able to see which patients they are assigned for and what are their conditions right now and also will be able to send their current condition to the doctor.
* Assistant will be able to add medicines and its companies and also will be able to set and get patient’s bill.

**CHAPTER 2: Background**

**2.1 Introduction**

Integration of documentation and knowledge based computerized services in “Digital Sheba” system is a tremendous means of helping health professionals in their daily practices and to improve the quality of services delivery.

The project "Digital Sheba" includes registration of patients, storing their details into the system, and also computerized billing in the pharmacy, and labs. The system will store patient’s records, drug inventory and dispensation as well as other relevant clinical information within the Hospital. The system will be used by the staff (Admin, Doctor, Nurse and Assistant) within the Hospital. It will process data speedily and accurately provide information when and where required. The System will be used to store data, produce reports and handle management inquiries. The interface is very user-friendly. The data are well protected for personal use and makes the data processing very fast. The purpose of the project entitled as “DIGITAL SHEBA” is to computerize the Front Office Management of Hospital to develop software which is user friendly, simple, fast, and cost – effective. It deals with the collection of patient’s information, diagnosis details, etc. Traditionally, it was done manually. The main function of the system is to register and store patient details and doctor details and retrieve these details as and when required, and also to manipulate these details meaningfully System input contains patient details, diagnosis details; while system output is to get these details on to the CRT screen.

Therefore, it is hoped that the system would provide a flexible and reliable management for the hospitals. This program is carefully designed to work in a standalone system or in a multi user environment.

**2.4 Related Work**

Our work plan of “Digital Sheba” doesn’t meet any kind of similar workings right now in our country. But we can see there many projects out their name Hospital Management system but they are implemented outside of our country. Also, one of our university name BRAC University has published about their project but there is no real-life implementation out there of it. Also, if we compare our idea with them, we can see they have added only patient and doctor as an actor but here in our project we have added 5 different types of actors and also, we have managed each of their role with sustainedly. Their imagined system only able to create doctor appointment and schedule the patient role but our system is able to visualize the patient condition, assigned doctor and nurse, medicine taking time etc.

Our work is attempting to create a web application, which provides a fast, easy to use interface for **“Digital Sheba”**. The main target of the work is to create platform independent. Web application to manage and distribute electronic documents just using Web Browser. The point is to create applications that do not need printable copies of documents.

They have worked for a small area and only for completing their university project but we have worked for implementing it to real world to solve problems of a patient that they face every day.

**2.3 Comparative Studies**

Comparative studies mean doing examinations between one or more developed systems with developed systems. Here we can find the similarities and differences between the system. These studies carry the most important part of any project.

It helps us to understand the uniqueness and help finding the proper solution of the current system. Here is some case study of our current system:

* **Case Study 1**: (Dashboard) This study defines dashboard as a display or visualization of highly important information which is used to fulfill necessary objectives, which is consolidated and displayed on a single screen so that all the necessary information can be monitored at once like how many patients are registers, how many doctors are in the hospital which medicines has been assigned for patients and so on. Dashboards are an information display which is predominantly visual that is used to monitor current scenarios at a glance so that a timely response can be given and hence are single-paged, glanceable visuals of data.
* **Case Study 2**: (Assign Nurse) This study defines that doctor will assign the specific nurse from the list for the specific patient and patient will get service from the specific nurse regularly and timely.
* **Case Study 3**: (Assign Doctor) This study defines that admin will assign the specific doctor from list for the specific patient and patient will get service from that specific doctor and also the assigned doctor will prescribe medicines for that patient and give his recent condition as a medical health status of him/her.
* **Case Study 4**: (Assign Assistant) This study defines that admin will assign assistant for a specific doctor and the doctor will get valuable data about patient from that assistant.
* **Case Study 5**: (Feedback & Reports) Here Admin will be able to see both feedbacks and reports from patient and doctor will be able to see only feedbacks from that patient.
* **Case Study 6**: (Assistant) This process for adding medicine and it’s information into the database and also check and set billing information of a patient.

**2.4 Scope and Features**

1. Information about Patients is done by just writing the patients name, age, gender. Whenever the Patient comes up his information is stored freshly.
2. Bills are generated by recording price for each facility provide to Patient on a separate sheet and at last they all are summed up.
3. Diagnosis information to patients is generally recorded on the document, which contains patient information. It is destroyed after some time period to decrease the paper load in the office.
4. Information about various diseases is not kept as any document. Doctors themselves do this job by remembering various medicines.

Hence, this platform will be so user-friendly for which patients will understand it easily without any hesitation. The application has a notification system for patients, which will notify them quickly when getting their medicine and other equipment. It will also have a feedback system implemented by which patient will be able to give feedback about nurses and which be allocated via admin and doctors directly.

**2.5 Challenges**

For developing a system, developers faced many problems and challenges to work on it. While we are developing this system, we also have faced some problems and challenges.

The challenges we have faced:

* **Professionalism:** As a junior developer we have some lack of professional experience. That’s why our work does not consist of 100% professionalism. But our work meets a grade solution able.
* **Technology Stack:** In our **“Digital Sheba”** project we are using different types of technology stacks like Django (Web Framework of Python), JavaScript’s, Bootstrap, Html and Css for front-end and back-end development. As a new developer those were challengeable for us. Those are the primary technologies of our system.
* **UI/UX Design:** Doing UI/UX design was also a challenge for us. Because we are new to doing this type of project. Actually, we enjoy doing this design challenge because it helps us to become more efficient and be more creative.
* **Docker:** As a new developer Docker concept was not so easy for us to use in our project. Cause making container and dockerize the project is so hard for first time as a new developer.
* **Deploy In VPS:** As a new developer deploying a project into a server is quite challengeable for us. Though we have deployed it correctly though.

**Chapter 3: Requirement Specification**

**3.1 Requirement Collection & Analysis**

Our proposed system will be available for patient, doctor, nurse, assistant and admin. This system provides different features for each user. There are totally five users in our system. The full system workflow of different users is described in bellow:

**Admin User:**

* Admin can edit and delete patient information only because patient will be registered by themselves.
* Admin can add, update and delete nurse information
* Admin can add, update and delete doctor information
* Admin can add, update and delete assistant information
* Also, admin will be able to assign doctor for a patient and can assign assistant for a doctor and also, he can update and delete that information.
* Admin will also have the ability to check patients reports and feedbacks every day and also, he has the ability to delete them.

**Doctor User:**

* Doctor can give prescription for a patient. Also, he can delete those prescription.
* Doctor can read patient recent condition and also will be able to delete.
* Doctor can assign nurse for a patient and also can delete them.
* Doctor can give report of a patient and delete them.
* Not only admin but also doctor can check patients’ feedbacks.

**Nurse User:**

* Nurse can get patients’ information about which medicine has been added for him and his other information that basically need for checkup.
* Nurse can add patients’ recent condition and which will be updated into doctors dashboard.

**Assistant User:**

* Assistant can add medicine company names and medicine. Here the main thing is to insert medicine information which will be provided to patients via doctor user.
* Assistant can set billing information by the patients’ usage report like which services patient used.
* Before set billing information Assistant will get bill from patient and update that patient have paid or not paid their bills.

**System User (Patient):**

* System User must be a registered user before using the system.
* He will be able to login with his registered credentials.
* He can see his current health status, assigned medicines and assigned doctors’ and nurses’.
* User will be able to give feedback and reports which will be managed by admin user and doctor user.

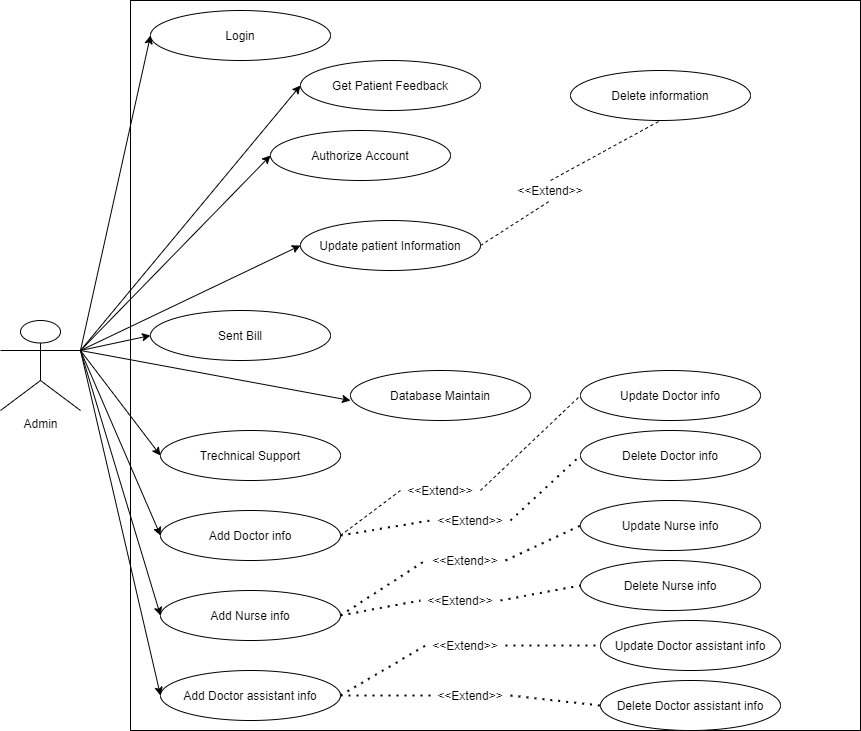
**3.2 Use Case Modeling & Design**

Use case model is an approach that is a combination of text and pictures in order to improve the understanding of requirements. A use case model describes the complete functionality of a system by identifying how everything that is outside the system interacts with it.

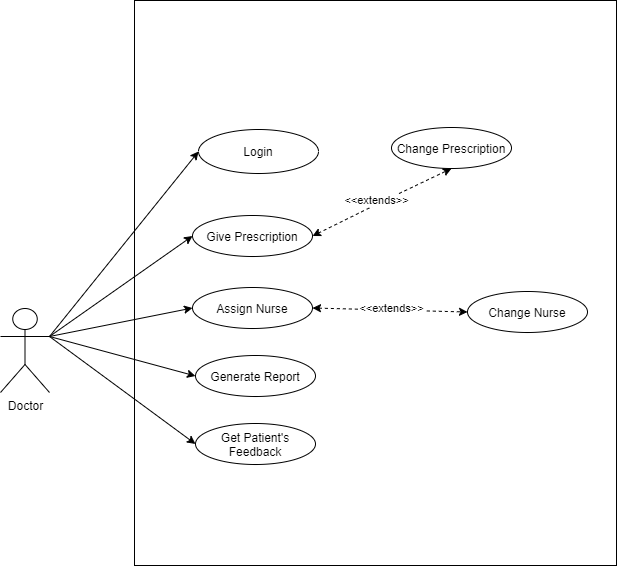
A Use Case Diagram is given below that relates to this application:

* Description – This project is a web application that manages a system of Enterprise Resource Planning
* Actors – It has 5 actors
  + Administration
  + Doctor
  + Nurse
  + Assistant
  + Patient

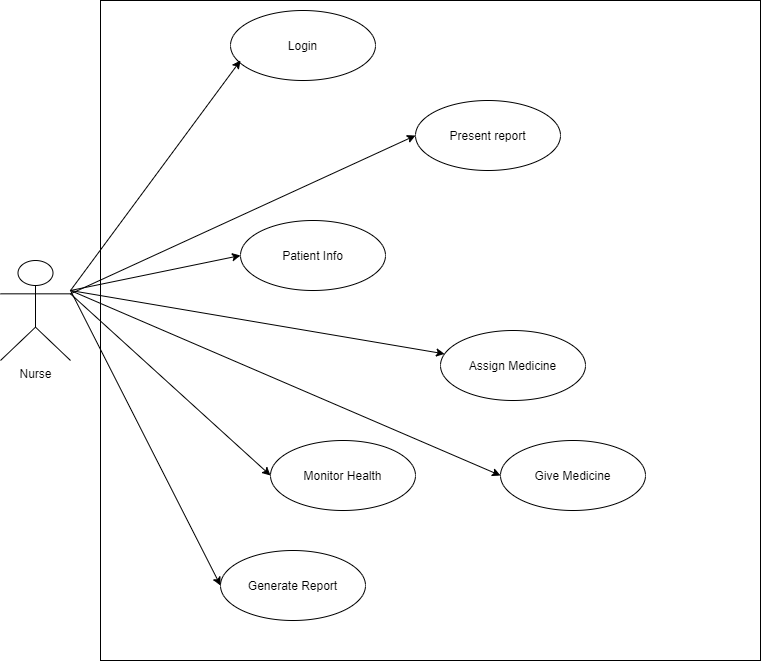
**3.2.1 Use Case Diagram for Admin**

****

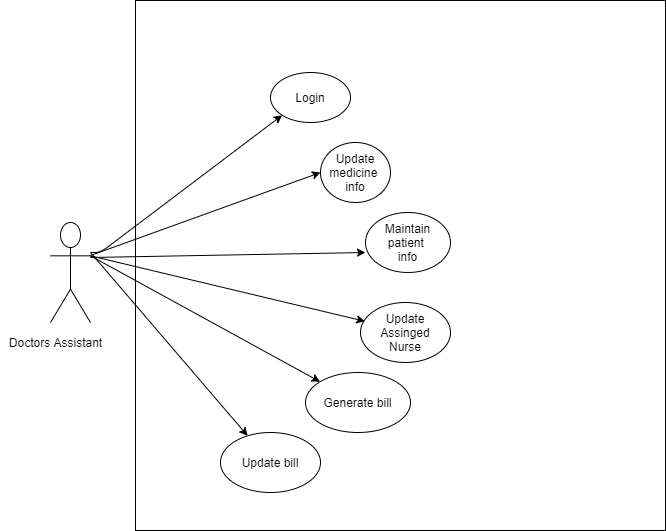
**3.2.2 Use Case Diagram for Doctor**

****

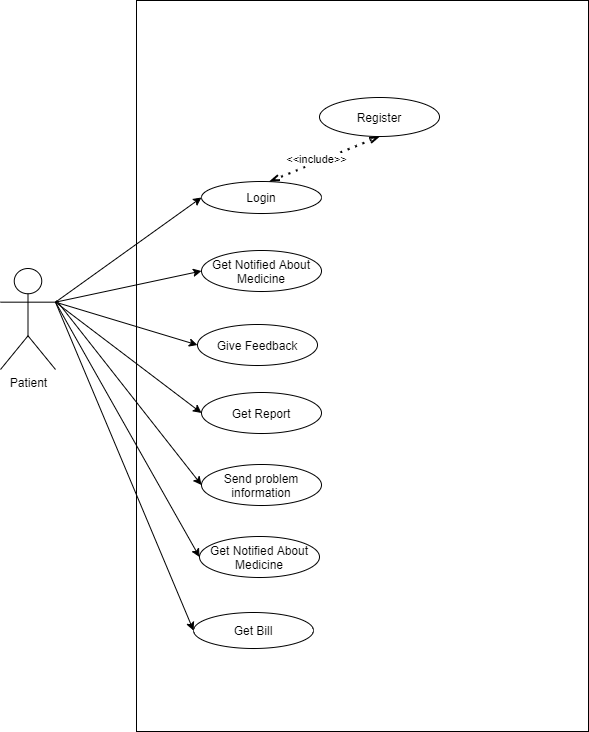
**3.2.3 Use case diagram for Nurse**

****

**3.2.4 Use Case Diagram For Assistant**

****

**3.2.5 Use Case Diagram For Patient**

****

**3.2.6 Use Case Description**

**USE CASE DESCRIPTION**

|  |
| --- |
| **Use Case [1]: Register** |
| **Description** |
| Patients must register first to access the online services of the system via website. |
| **Trigger** |
| Someone enters the website and clicks register button |
| **Actors** |
| Patient |
| **Precondition** |
| Patient should open mobile application with active internet connection and should not have an existing account |
| **Post Condition** |
| Patients accessing online can use all services that has been made for them after registration |
| **Success Scenario** |
| A patient without an existing account accesses the website and registers himself/herself filling the registration form appropriately. |

|  |
| --- |
| **Use Case [2]: Login** |
| **Description** |
| Patients must be registered before logged into it and other user accounts must be created via admin |
| **Trigger** |
| Someone enters the website and clicks login button |
| **Actors** |
| Patient, Doctor, Doctor’s Assistant, Nurse, Admin |
| **Precondition** |
| Only patient will be accessible through mobile application and other users have to be connected through web and active internet connection will be required for it. |
| **Post Condition** |
| After successful login, all users can use all online services of the system which have been provided for them. |
| **Success Scenario** |
| Every user with an existing account accesses the website and logins with accurate credentials |

|  |
| --- |
| **Use Case [3]: Get Notified About Medicine** |
| **Description** |
| Patients will be get notified every time when it’s time to take medicine also a nurse will be there to check them first before taking the medicine. |
| **Trigger** |
| Someone will check the notification |
| **Actors** |
| Patient |
| **Preconditions** |
| 1. Patient must have an existing account. 2. Patient or his family will visit that mobile application to check the notification with active internet connection. |
| **Post Conditions** |
| Patient can check notification about which medicine have been assigned for that time. |
| **Success Scenario** |
| Someone accesses the system and get notification |

|  |
| --- |
| **Use Case [4]: Give Feedback** |
| **Description** |
| Patients will be able to give feedback about nurse or assistant which has assigned for him/her if he/she feel that the services are not enough or not properly doing. |
| **Trigger** |
| Someone will start typing the feedback and click submit button. |
| **Actors** |
| Patient |
| **Preconditions** |
| 1. Patient must have an existing account. 2. Patient or his family will visit from mobile application and login into it to give feedback |
| **Post Conditions** |
| Patient will give feedback to doctor |
| **Success Scenario** |
| Someone accesses the system and write on feedback section and submit it. |

|  |
| --- |
| **Use Case [5]: Send Problem Info** |
| **Description** |
| Patient will be able to send his problem if he/she face any and that will be another section from feedback. |
| **Trigger** |
| Someone will start typing the problem and click submit button. |
| **Actors** |
| Patient |
| **Preconditions** |
| 1. Patient must have an existing account. 2. Patient or his family will visit from mobile application and login into it to send problem |
| **Post Conditions** |
| Patient will send information about his problem |
| **Success Scenario** |
| Someone accesses the system and write on problem information section and submit it successfully. |

|  |
| --- |
| **Use Case [6]: Get Patient Present Report** |
| **Description** |
| Patient will be able to see his present condition. |
| **Trigger** |
| Someone will visit and click report button |
| **Actors** |
| Patient |
| **Preconditions** |
| 1. Patient must have an existing account. 2. Patient or his family will visit from mobile application and login into it to see the report |
| **Post Conditions** |
| No post condition is applicable for it. |
| **Success Scenario** |
| Someone accesses the system and see the recent condition of patient. |

|  |
| --- |
| **Use Case [7]: Get Billing Info** |
| **Description** |
| Patient will be able to check total billing information after he released from the hospital. |
| **Trigger** |
| Someone will click to Payment Button |
| **Actors** |
| Patient |
| **Preconditions** |
| 1. Patient must have an existing account. 2. Patient or his family will visit from mobile application and login into it to check the bill |
| **Post Conditions** |
| Patient will check total bill after release |
| **Success Scenario** |
| Someone accesses the system and check total billing information successfully. |

|  |
| --- |
| **Use Case [8]: Give Prescription** |
| **Description** |
| Doctor will give prescription |
| **Trigger** |
| Someone will start typing and fill the fields and click submit button. |
| **Actors** |
| Doctor |
| **Preconditions** |
| 1. Doctor must have an existing account. 2. Doctor will visit the webpage provided for him. |
| **Post Conditions** |
| Doctor will give medicine information |
| **Success Scenario** |
| Someone accesses the system and give medicine information successfully |

|  |
| --- |
| **Use Case [9]: Change Prescription** |
| **Description** |
| Doctor will be able to change his prescription also if needed |
| **Trigger** |
| Someone will start typing and fill the fields and click submit button |
| **Actors** |
| Doctor |
| **Preconditions** |
| 1. Doctor must have an existing account. 2. Doctor will visit the webpage provided for him. |
| **Post Conditions** |
| Doctor will give medicine information |
| **Success Scenario** |
| Someone accesses the system and change medicine information successfully |

|  |
| --- |
| **Use Case [10]: Assign Nurse** |
| **Description** |
| Doctor will be able to assign nurse for each patient |
| **Trigger** |
| Someone will start choosing and click submit button |
| **Actors** |
| Doctor |
| **Preconditions** |
| 1. Doctor must have an existing account. 2. Doctor will visit the webpage provided for him. |
| **Post Conditions** |
| Doctor will watch nurse list and choose one for each patient |
| **Success Scenario** |
| Someone accesses the system and choose nurses for the patient and assign them successfully |

|  |
| --- |
| **Use Case [11]: Change Nurse** |
| **Description** |
| Doctor will be able to change nurse according to a patients feedback if needed |
| **Trigger** |
| Someone will start choosing and click submit button |
| **Actors** |
| Doctor |
| **Preconditions** |
| 1. Doctor must have an existing account. 2. Doctor will visit the webpage provided for him. |
| **Post Conditions** |
| Doctor will watch nurse list and choose one for each patient |
| **Success Scenario** |
| Someone accesses the system and choose nurses for the patient and assign them successfully |

|  |
| --- |
| **Use Case [12]: Generate Report** |
| **Description** |
| Doctor will be able to generate report after watching recent condition of a patient |
| **Trigger** |
| Someone will start typing and filling the fields and click submit. |
| **Actors** |
| Doctor |
| **Preconditions** |
| 1. Doctor must have an existing account. 2. Doctor will visit the webpage provided for him. |
| **Post Conditions** |
| Doctor will complete typing all of those fields and submit it |
| **Success Scenario** |
| Someone accesses the system with active internet connection and fill all of those fields and submit it from the webpage which has been provided for him. |

|  |
| --- |
| **Use Case [13]: Get Patients Feedback** |
| **Description** |
| Doctor and Admin will be able to get patients feedback about the service they are getting them. |
| **Trigger** |
| Someone will open page and see the list |
| **Actors** |
| Doctor, Admin |
| **Preconditions** |
| 1. Doctor and Admin must have an existing account. 2. Doctor and Admin will visit the webpage provided for him. |
| **Post Conditions** |
| No post condition is required for this |
| **Success Scenario** |
| Someone accesses the system and start watching those reports |

|  |
| --- |
| **Use Case [14]: Get Patient present report** |
| **Description** |
| Nurses will be able to see patients present report which has been provided from doctor |
| **Trigger** |
| Someone will visit and watch the report |
| **Actors** |
| Patient, Nurse |
| **Precondition** |
| 1. Nurse must have an existing account 2. Nurse will visit the webpage with active internet connection |
| **Post Condition** |
| No post condition is required for this action |
| **Success Scenario** |
| Someone access the provided system and start watching the reports |

|  |
| --- |
| **Use Case [15]: Get patient info** |
| **Description** |
| Nurses will be able to see patient’s information which has been provided by patients. |
| **Trigger** |
| Someone will visit and watch patient info |
| **Actors** |
| Nurse |
| **Precondition** |
| 1. Nurse must have an existing account 2. Nurse will visit the webpage with active internet connection |
| **Post Condition** |
| No post condition is required for this action |
| **Success Scenario** |
| Someone access the provided system and start watching patient’s information |

|  |
| --- |
| **Use Case [16]: Allocate Assigned medicine** |
| **Description** |
| Nurses will be able to allocate which medicine has been provided for his/her patient for which he/she is assigned for |
| **Trigger** |
| Someone will visit and watch medicine names |
| **Actors** |
| Nurse |
| **Preconditions** |
| 1. Nurse must have an existing account 2. Nurse will visit the webpage with active internet connection |
| **Post Conditions** |
| No post condition is required for this action |
| **Success Scenario** |
| Someone access the provided system and start checking medicine names. |

|  |
| --- |
| **Use Case [17]: Monitor Patient Health** |
| **Description** |
| Nurses will monitor patient heath. This action will be done directly. So here the apps will do nothing. |
| **Trigger** |
| No action trigger required for this field. |
| **Actors** |
| Nurse |
| **Preconditions** |
| No precondition required for this. |
| **Post Conditions** |
| No post condition required for this. |
| **Success Scenario** |
| Someone will check current condition of a patient by directly going to them. |

|  |
| --- |
| **Use Case [18]: Update Medicine Info** |
| **Description** |
| Assistant will update medicine info after gathering it from doctor |
| **Trigger** |
| Someone will visit and start typing and click submit button |
| **Actors** |
| Doctor’s Assistant |
| **Preconditions** |
| 1. Doctor’s assistant must have an existing account. 2. Assistant will visit webpage and login into it. |
| **Post Conditions** |
| Assistant will update medicine info |
| **Success Scenario** |
| Someone will visit the webpage and start updating medicine info |

|  |
| --- |
| **Use Case [19]: Maintain patient info** |
| **Description** |
| Assistant will be able to see and maintain patient information when needed |
| **Trigger** |
| Someone will visit and start checking |
| **Actors** |
| Doctor’s Assistant |
| **Preconditions** |
| 1. Doctor’s assistant must have an existing account. 2. Assistant will visit webpage and login into it. |
| **Post Conditions** |
| Assistant will check each and every details of a patient |
| **Success Scenario** |
| Someone will check patient information and give response to doctor |

|  |
| --- |
| **Use Case [20]: Update Assigned Nurse** |
| **Description** |
| Assistant will be able to update assigned nurse after getting the information from doctor. |
| **Trigger** |
| Someone will visit and start typing new information and click submit button |
| **Actors** |
| Doctor’s Assistant |
| **Preconditions** |
| 1. Doctor’s assistant must have an existing account. 2. Assistant will visit webpage and login into it. |
| **Post Conditions** |
| Assistant will update newly assigned nurse for the patient |
| **Success Scenario** |
| Someone will visit the website and update patient information |

|  |
| --- |
| **Use Case [21]: Generate / Update Bill** |
| **Description** |
| Assistant will be generate or update billing information according to service that patients get every day. |
| **Trigger** |
| Someone will visit and start typing new information and click submit button |
| **Actors** |
| Doctor’s Assistant |
| **Preconditions** |
| 1. Doctor’s assistant must have an existing account. 2. Assistant will visit webpage and login into it. |
| **Post Conditions** |
| Assistant will generate and update billing information |
| **Success Scenario** |
| Someone will visit the website and generate or update billing information everyday into the website |

|  |
| --- |
| **Use Case [22]: Authorize Patient Account** |
| **Description** |
| Admin will authorize patient information after their registration to the site. |
| **Trigger** |
| Someone will check patient information and click submit button |
| **Actors** |
| Admin |
| **Preconditions** |
| 1. Admin must have an existing account. 2. Admin will visit webpage and login into it. |
| **Post Conditions** |
| Admin will check patient information and authorize them. |
| **Success Scenario** |
| Someone will visit the website and check patient information each and every details and authorize them. |

|  |
| --- |
| **Use Case [23]: Update patient information** |
| **Description** |
| Admin will be able to update patient information if needed. |
| **Trigger** |
| Someone will visit and start typing new information and click submit button |
| **Actors** |
| Admin |
| **Preconditions** |
| 1. Admin must have an existing account. 2. Admin will visit webpage and login into it. |
| **Post Conditions** |
| Admin will check and update patient information |
| **Success Scenario** |
| Someone will visit and update patient information on demand. |

|  |
| --- |
| **Use Case [24]: Delete Patient Information** |
| **Description** |
| Admin will be able to delete patient information when they will be released from the hospital |
| **Trigger** |
| Someone will visit the website and click delete button |
| **Actors** |
| Admin |
| **Preconditions** |
| 1. Admin must have an existing account. 2. Admin will visit webpage and login into it. |
| **Post Conditions** |
| Admin will delete patient information |
| **Success Scenario** |
| Someone will visit the website and delete patient information. Though deleting information is not a good practice for a system. They basically store them into another portion named (Trash) |

|  |
| --- |
| **Use Case [25]: Set Bill To Patient** |
| **Description** |
| Admin will be able to update billing information which will be generated from doctor’s assistant to patient profile. |
| **Trigger** |
| Someone will visit and start typing new information and click submit button |
| **Actors** |
| Assistant |
| **Preconditions** |
| 1. Assistant must have an existing account. 2. Assistant will visit webpage and login into it. |
| **Post Conditions** |
| Admin will update billing information to patient profile |
| **Success Scenario** |
| Someone will visit the website and update billing information to patient profile every day. |

|  |
| --- |
| **Use Case [26]: Add/ Update/ Delete Doctor** |
| **Description** |
| Admin will be able to manage each and every information of a doctor. Here he will be able to add a new doctor, update that doctor’s information or delete that doctor’s information if needed. |
| **Trigger** |
| Someone will visit and start typing and click submit button or delete button when it needed. |
| **Actors** |
| Admin |
| **Preconditions** |
| 1. Admin must have an existing account. 2. Admin will visit webpage and login into it. |
| **Post Conditions** |
| Admin will manage doctor’s information. |
| **Success Scenario** |
| Someone will visit the website and do those crud operation when it’s needed to do. |

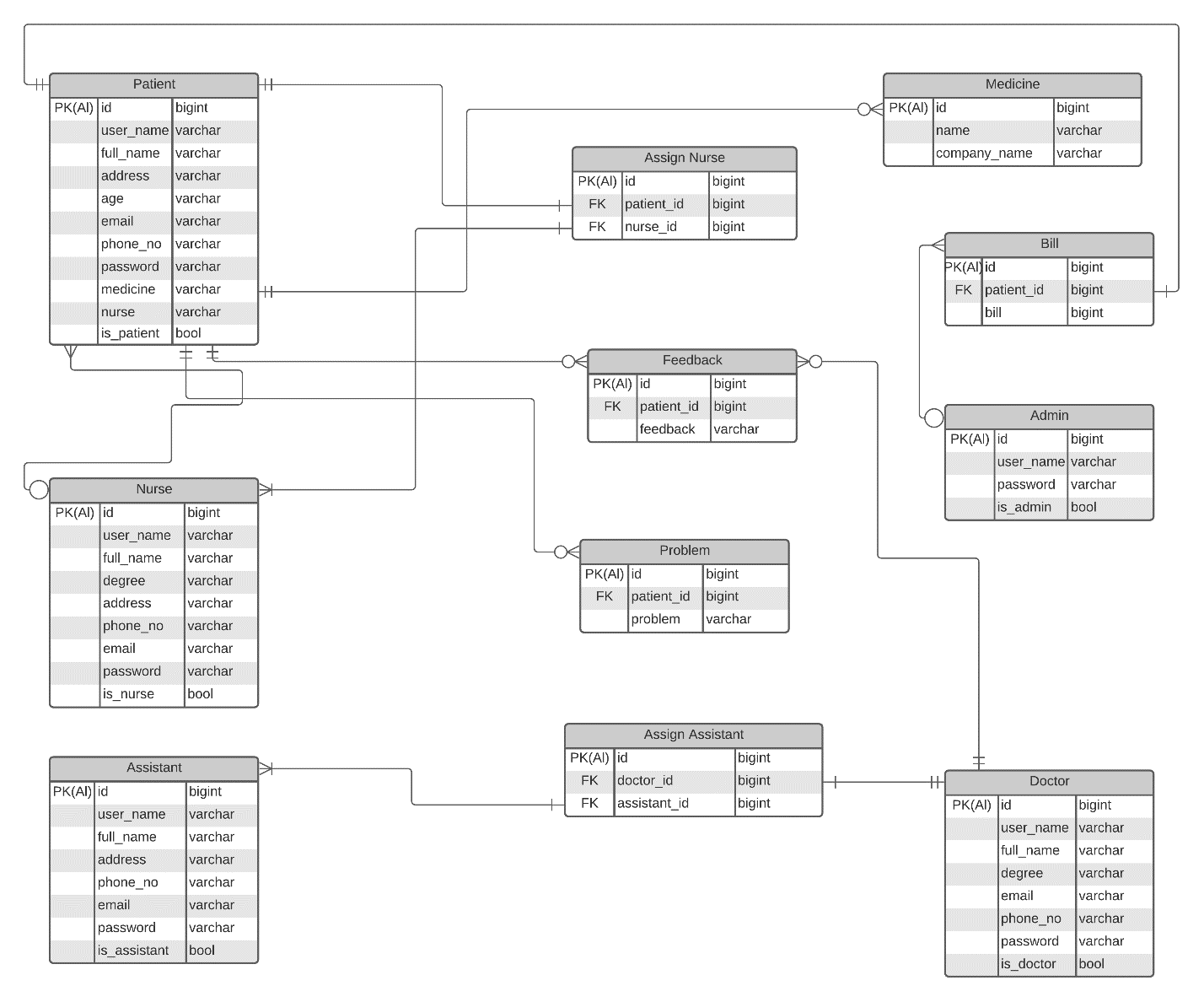
|  |
| --- |
| **Use Case [27]: Add/ Update/ Delete Nurse** |
| **Description** |
| Admin will be able to manage each and every information of a Nurse. Here he will be able to add a new Nurse, update that Nurse’s information or delete that Nurse’s information if needed. |
| **Trigger** |
| Someone will visit and start typing and click submit button or delete button when it needed. |
| **Actors** |
| Admin |
| **Preconditions** |
| 1. Admin must have an existing account. 2. Admin will visit webpage and login into it. |
| **Post Conditions** |
| Admin will manage nurse’s information. |
| **Success Scenario** |
| Someone will visit the website and do those crud operation when it’s needed to do. |

|  |
| --- |
| **Use Case [28]: Add/ Update/ Delete Doctor’s Assistant** |
| **Description** |
| Admin will be able to manage each and every information of a assistant. Here he will be able to add a new Assistant, update that assistant’s information or delete that assistant’s information if needed. |
| **Trigger** |
| Someone will visit and start typing and click submit button or delete button when it needed. |
| **Actors** |
| Admin |
| **Preconditions** |
| 1. Admin must have an existing account. 2. Admin will visit webpage and login into it. |
| **Post Conditions** |
| Admin will manage assistant’s information. |
| **Success Scenario** |
| Someone will visit the website and do those crud operations when it’s needed to do. |

|  |
| --- |
| **Use Case [29]: Database Maintenance** |
| **Description** |
| Admin will be able to see and maintain database information everyday. |
| **Trigger** |
| Someone will visit database and check for vulnerability if needed. |
| **Actors** |
| Admin |
| **Preconditions** |
| 1. Admin must have an existing account. 2. Admin will visit webpage and login into it. |
| **Post Conditions** |
| Admin will manage database. |
| **Success Scenario** |
| Someone will visit into database section and check for security breach. |

|  |
| --- |
| **Use Case [30]: Technical Support** |
| **Description** |
| Admin will be able to give any kind of technical support to any kind of users. |
| **Trigger** |
| No triggers Required |
| **Actors** |
| Admin |
| **Preconditions** |
| 1. Admin must have an existing account. 2. Admin will visit webpage and login into it. |
| **Post Conditions** |
| Admin will give technical support. |
| **Success Scenario** |
| Someone will visit the site and watch support tickets. |

**3.4 Logical Model Data**

****

**3.5 Design Requirements**

**Performance**

Our system is very user friendly and it is fast enough to use. It response very fast to every user in a target operation. Performance is about timing which starts with a request to the system and process, the arriving request to generate a response. We measure the performance in terms of response time and the total process.

**Reliability**

The proposed system will be reliable in terms of stability and security. The system has been hosted on cloud (Digital Ocean) service.

**Security**

The system will never expose any personal information of any users and didn’t collect any personal information from its own users. Users data will be encrypted in the database.

**Portability**

Web Application and Cloud services make the system portable then users can use this system from anywhere.

**Maintainability**

Maintainability defines the time required for a solution or its component to be fixed, changed to increase performance or other qualities, or adapted to a changing environment. The system is very easy to maintain because of our easy and great user interface. The powerful backend makes the data more maintainable.

**Usability**

Users can use this system by logging in. Their account must be created by the end user. The user interface of the system should be in a new design sense for understanding.

**Availability**

The cloud service makes our web application available online anytime from anywhere around the world. Different applications (Web, Mobile) will make our user access to the data anytime.

**Chapter 4: Design Specification**

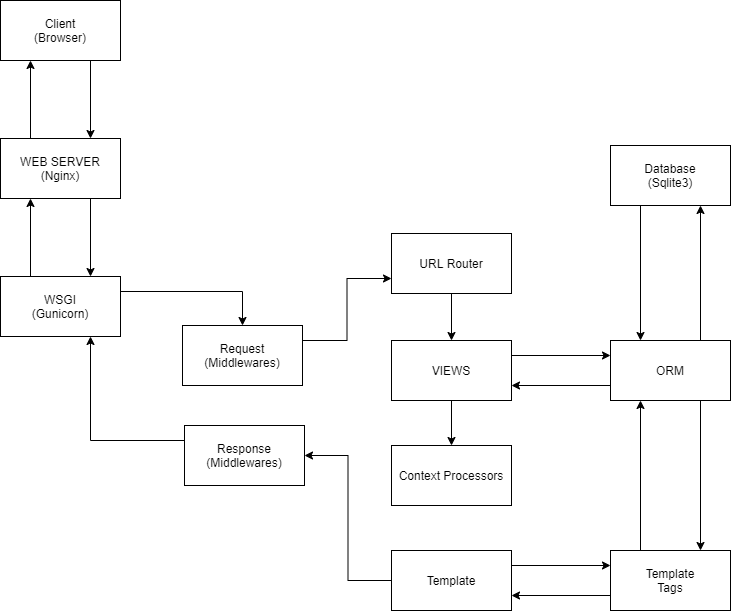
**4.1 Front-End Design**

In our Digital Sheba System, we tried to have maximum level integration of User interface, User experience & Performance. So, we come up with a solution to have the system so much lightweight & simple front-end design for the better user experience. Though maximum of our system is covered up with dashboard only because maximum actor will act with Dashboard. Only Patient will be not a dashboard user here. Our front-end design will be like as mentioned below with the all features.

* **Login/Logout:**
  + **Login:** The host server shall contain definitions for the user names, passwords & access roles e.g., administrator role, doctor role, nurse role, assistant role and patient role. The definition shall be local to the host server only and shall be displayed at the user location. The login shall succeed or fail. In case of a failure, a default authentication page shall display. The login shall be capable of cancellation. If cancelled the Home page or Dashboard page shall be displayed. A login failure shall redisplay the login method with all field blank.
  + **Logout:** After the successful login and the login page will not be accessible via any user. They will be redirected to Home page or Dashboard page and each and every page will carry a logout control option.
* **Dashboard:** I have already mentioned that our maximum user interface is dashboard and maximum actor role act is using dashboard. After login a user is presented with an interactive page which will contain many information like how many patients are admitted, how many feedbacks and reports are there, how many doctors’ and nurses are there to give services. Also, every actor who have control of dashboard will get another control service and they will make their own actions here. Like Admin have user control system where we have provided patient information control, doctor information control, nurse information control, assistant information control, assigned doctor and assistant information control in Control Info View. Also, Admin have got checking feedback and report views. Here doctor have got Doc Control where he has got Give prescription views, patient health check views, assign nurse views, give report views and getting feedback views. Nurse have got Get patient information views and Patient Report views where nurse will give patient recent condition to doctors and lastly Assistant actor have got Medicine Company and Medicine views where he will add medicine company name and medicine name and he will be able to set billing information and get bill from patient through the dashboard.
* **Services:** Patient will get main services after logged in. He will be able to see his recent Condition and also, he will be able to see which medicine has been assigned for him in his profile page. In medicine page he will see what is the time of taking medicine. He will also see which doctor and nurses has be assigned for him/her and how much amount he/she have to pay after released.

**4.2 Back-End Design**

Our application follows modern application structure which is Universal Application that enables our front-end pages and assets to be rendered on server-side. The application consists of two parts – Backend and Frontend. When a user sends request to the server, it primarily gets handled by Nginx Webserver. Nginx then redirects the request to WSGI (Gunicorn). Now Gunicorn sends a request to the middleware which hits the URL router and then send request to the views. Then views check the context processor to process data using ORM and hit the database and then return the rendered template to the middleware again to check the authentication and send the data and other components to Nginx. In the last step, Nginx will send the page as a response to the user and the user will see his desired page.



**Django:** Django is a high-level web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of web development, so we can focus on writing our app without needing to reinvent the wheel and also, it’s free and open source. Django is ridiculously fast and it was designed to help developers take applications from concept to completion as quickly as possible. It is reassuringly secure and it takes security seriously and helps developers avoid many common security mistakes and it’s exceedingly scalable and in some of the busiest sites on the web leverage Django’s ability to quickly and flexibly scale to meet the heaviest traffic demands. Finally, it’s incredibly versatile and also companies, organizations and governments have used Django to build all sorts of things from content management system to social networks to scientific computing platforms.

Authentication Best Practices:

Primary caution has been taken to ensure that leaking of information does not harm user authentication credentials. With that in mind, we have used Django authentication system and encryption algorithm name SHA512 to encrypt user’s password.

**4.3 Interaction Design and UX**

Course Tracking Management System have simple & most attractive interaction design. The user can find it very much easy to operate & use. The design is trendy fresh flat UI. A user can easily find what they need with simple some clicks. User can view & do the work for the System with great experience.

The User Interface is minimalistic. The User Experience is built with some criteria. The usability, accessibility and pleasure provide in the interaction with the user is totally satisfactory. There is no such bug or abominable task or interaction found while using or interaction with the system. The system interaction design, visual design, information architecture, user research and other disciplines, and is concerned with all facts of the overall experience delivered to users.

The most convenient feature of our developed system is responsiveness. The user interface is flexible with all type of display resolution even with apple’s retina display. The system is mobile ready on android or iPhone-based browser.

**4.4 Implementation Requirements**

**Technical Specification Requirements**

The system will be developed using the following technology

1. **Web Development Technology**

**Operating System:** Ubuntu 20.04

**Web Server:** Nginx

**Container:** Docker

**Application Architecture:** Universal Application (UA)

**Language Platform:** Python 3.6.10, JavaScript

**Backend Framework:** Django 3.1.2

**Frontend Framework:** Django Template Engine

**UI Component:** Bootstrap

**Database Server:** Sqlite3

**IDE:** Visual Studio Code by Microsoft

1. **Web Server Configuration (VPS):**

**Processor:** 1 VCPU (Virtual Central Processing unit)

**RAM:** 1 GB Ram

**Storage:** 25 GB

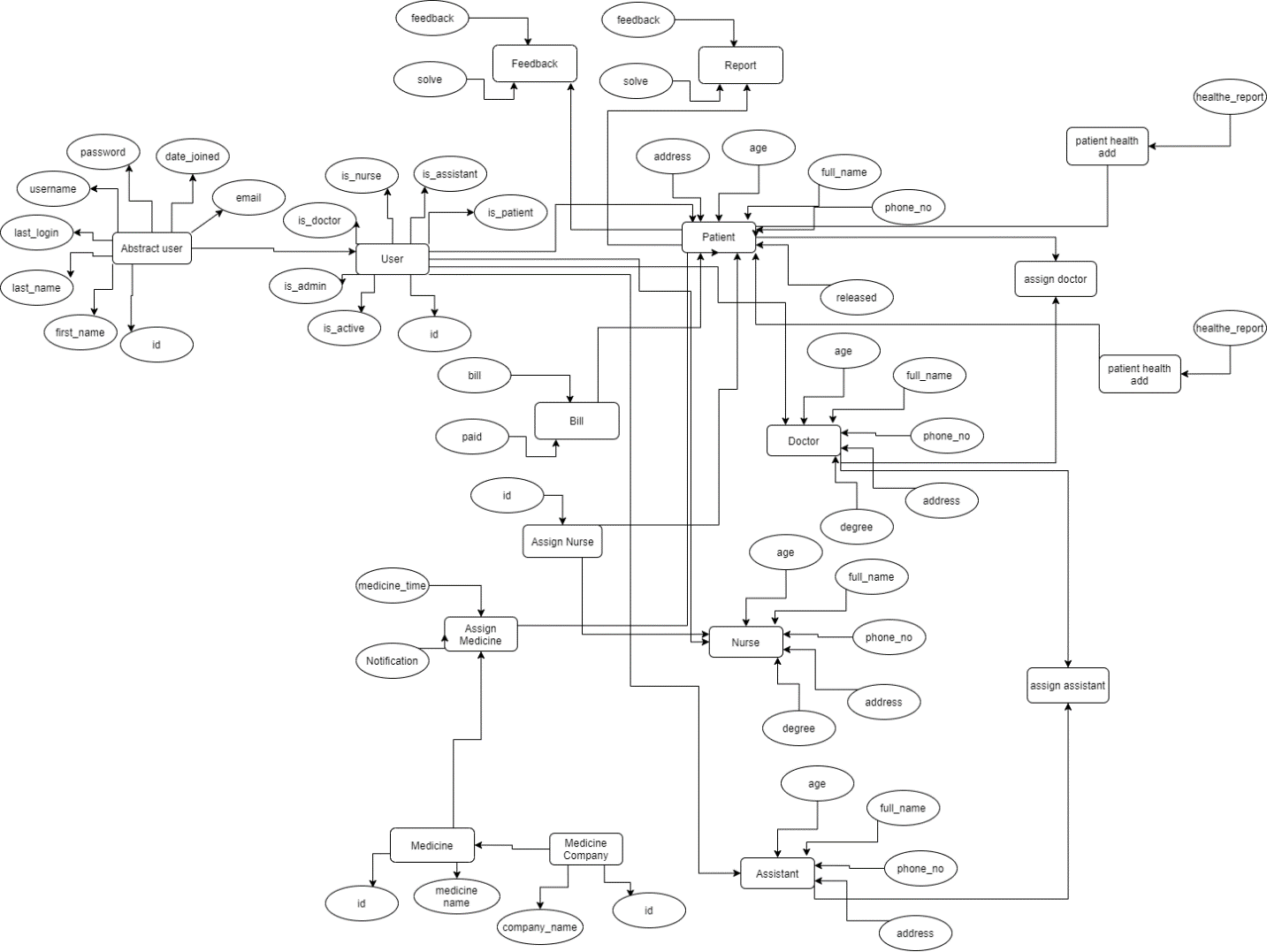
**Transfer Rate:** 1 TB Monthly Transfer(Bandwidth for better transfer of the files for the user)

1. **Domain Name Server:** One registered Domain with SSL Enabled SSL(Secure Sockets Layer) is the standard security technology for establishing an encrypted link between a web server and a browser. This link ensures that all data passed between the web sever and browsers remain private and integral.

**CHAPTER 5: Implementation and Testing**

**5.1 Implementation of Database**

For our developed system we follow the schema and build up an Entity Relationship Diagram for implementation of the database.



**5.2 Implementation of Front-end**

The developed system has met a minimalistic flat design & ultimate easy user understandable design solution. So, the UI & UX is too much easy for use & interaction with. But we also do not compromise the Design for the optimization. The system has a great loop to incorporate with. The figures, given bellow are Front-End Design of Implementation of the developed system.

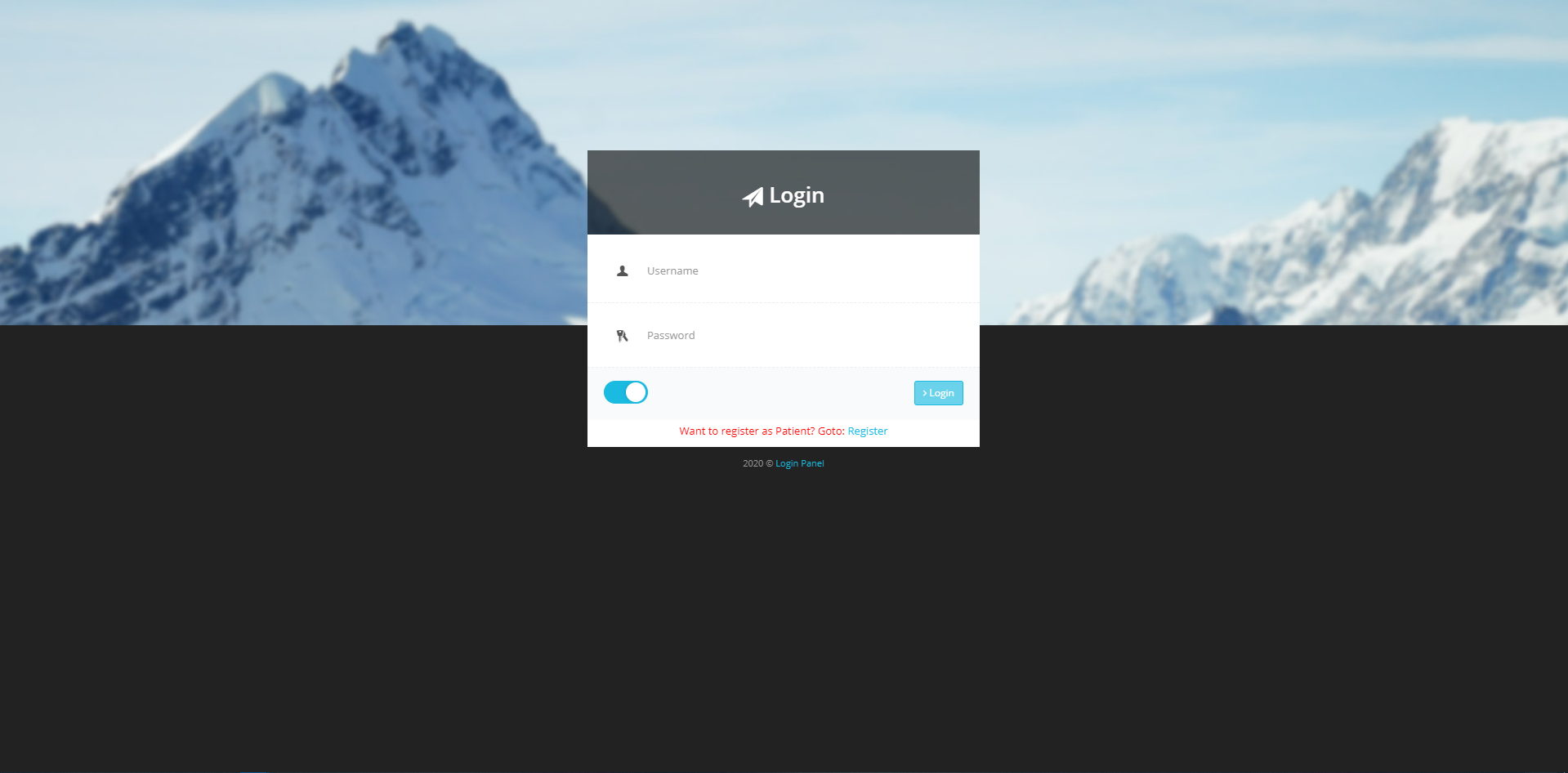


Figure 5.2.1

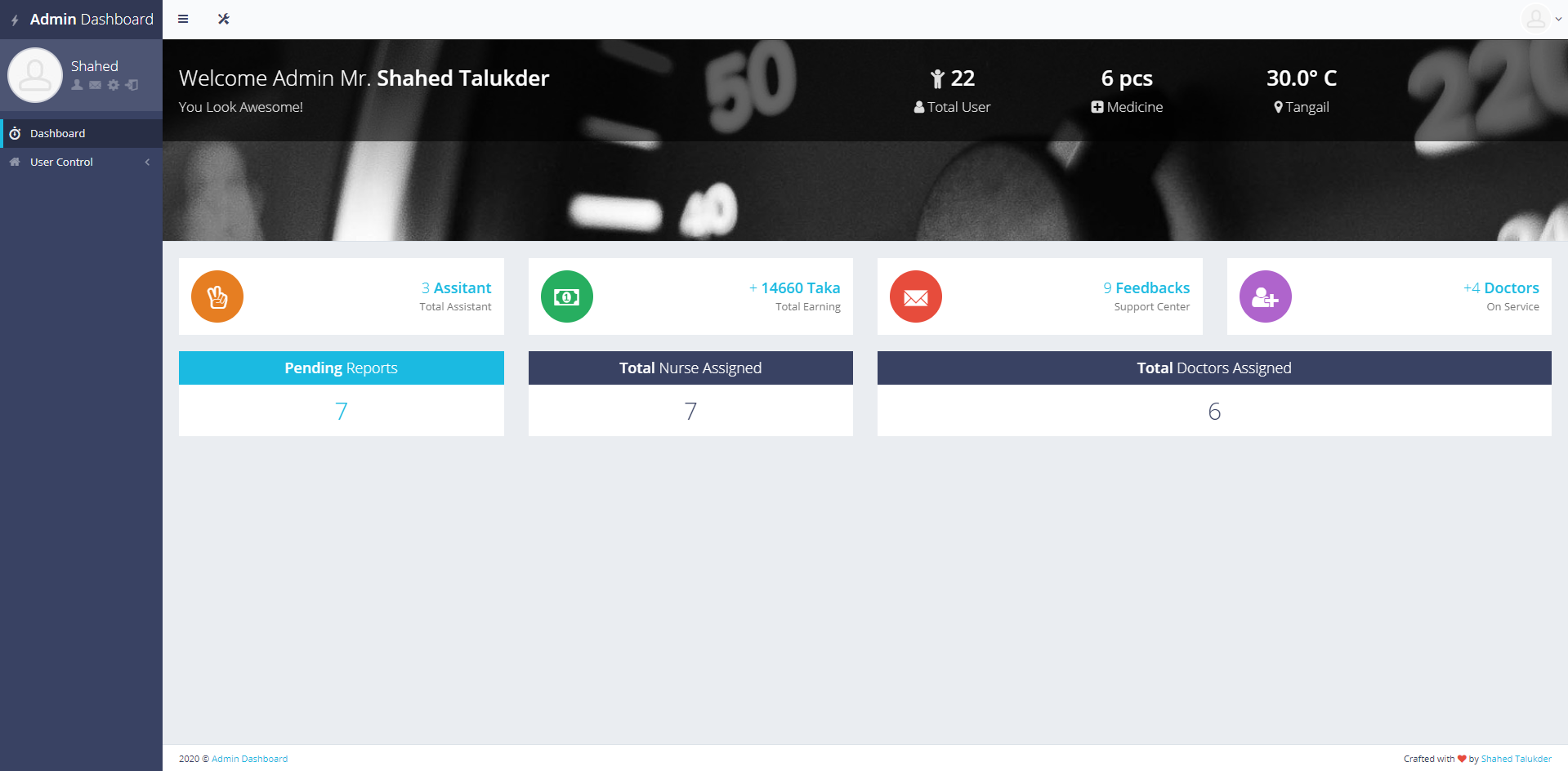


Figure 5.2.2

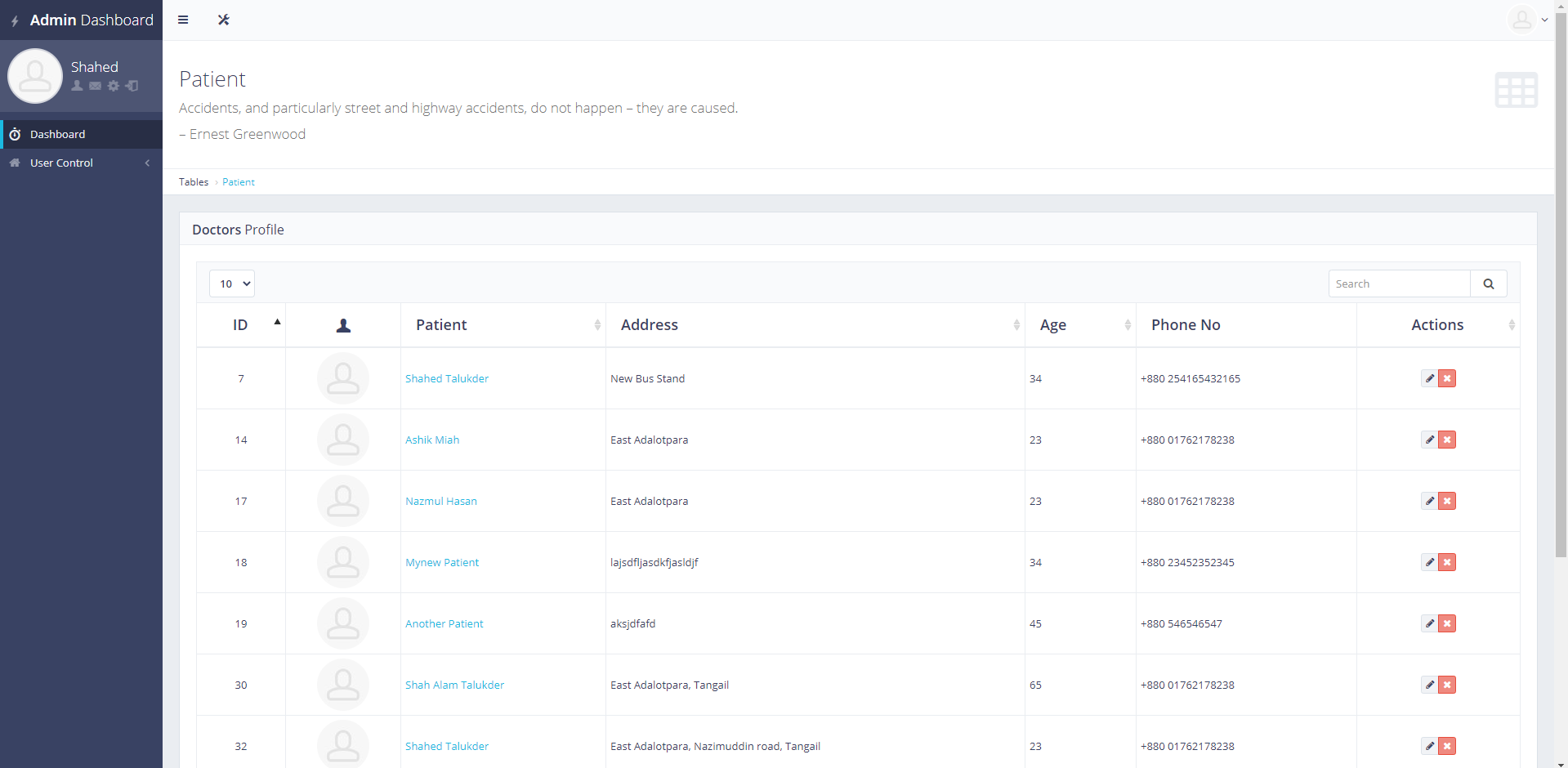


Figure 5.2.3

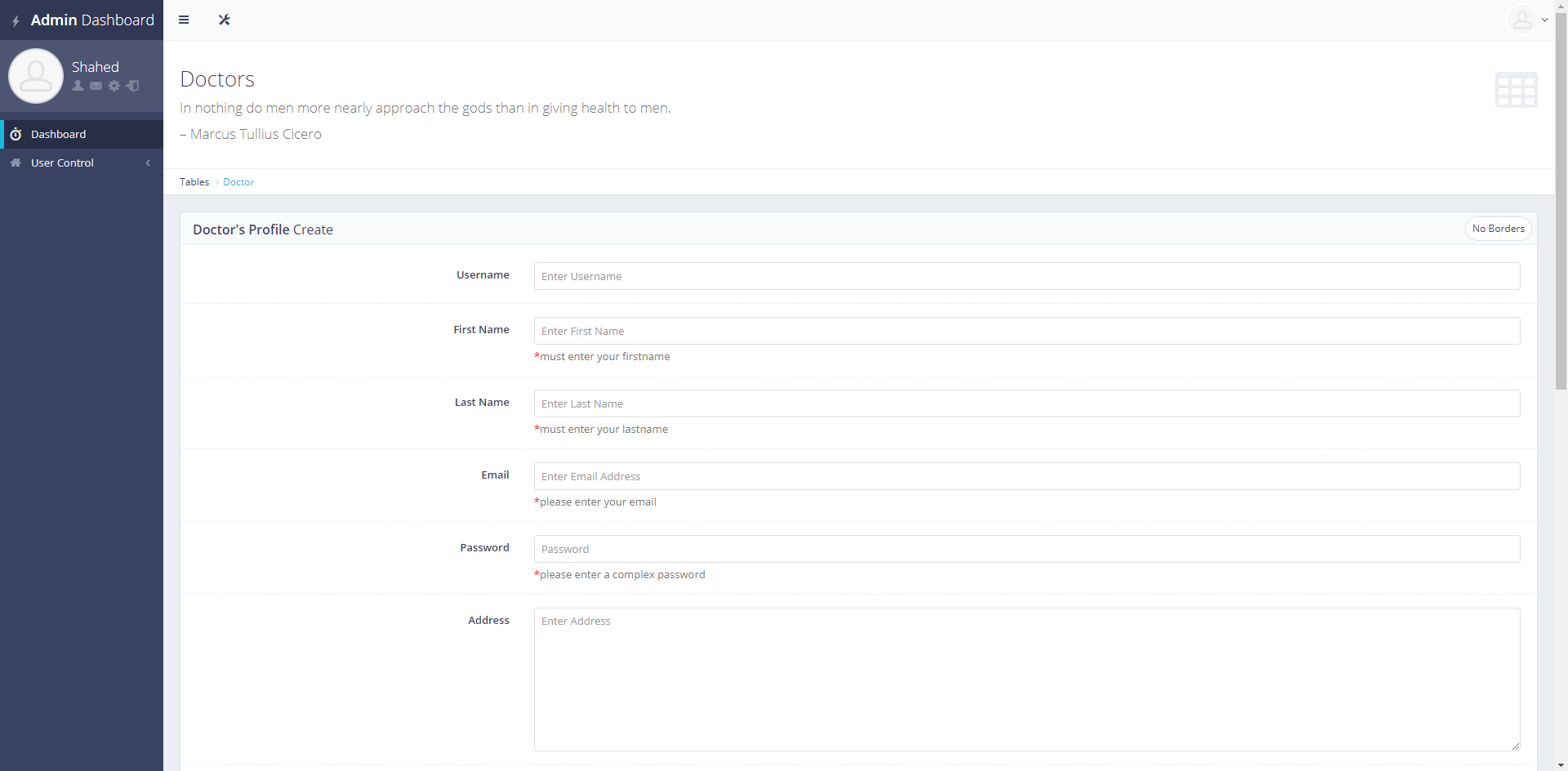


Figure 5.2.4

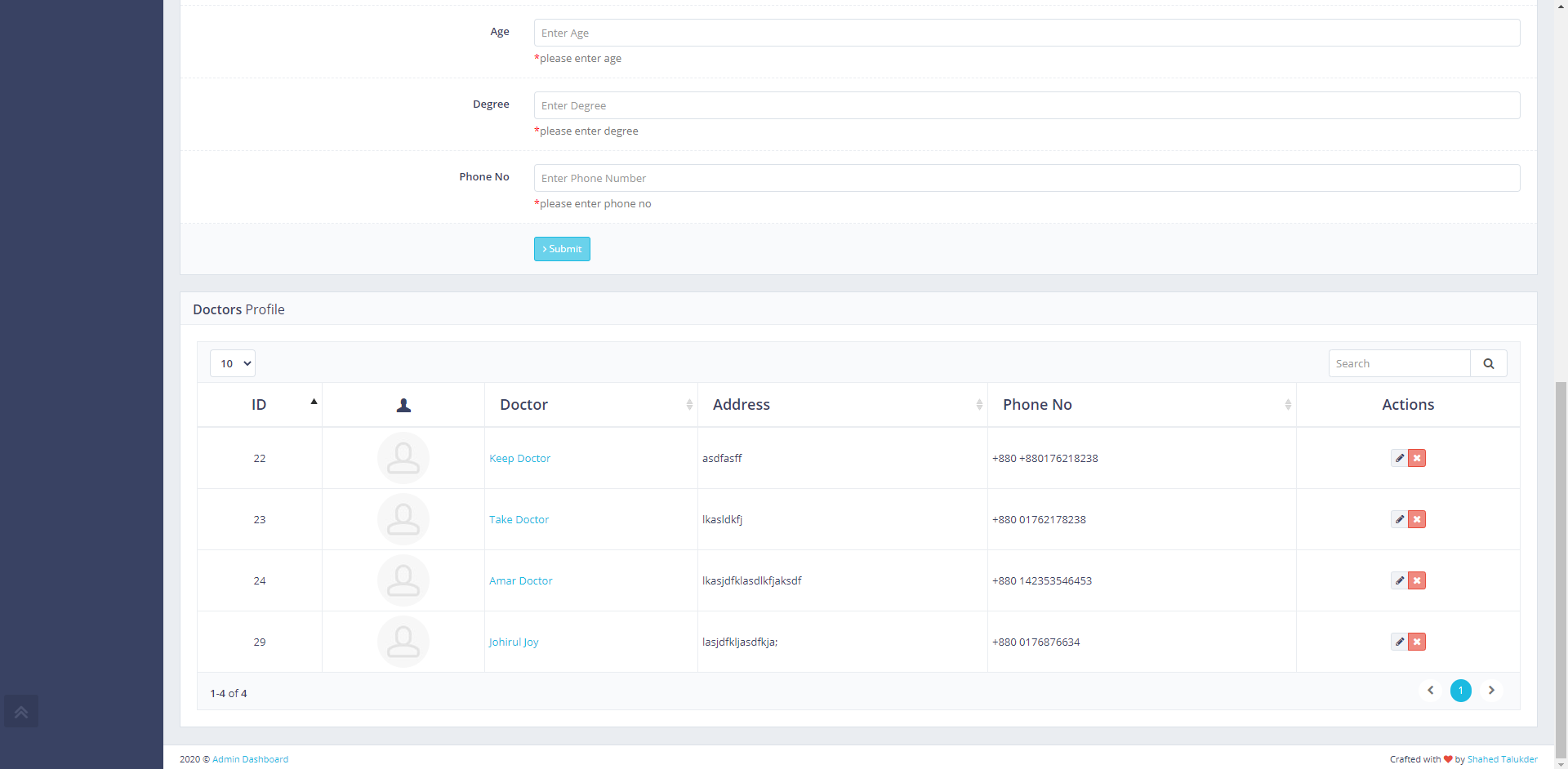


Figure 5.2.5

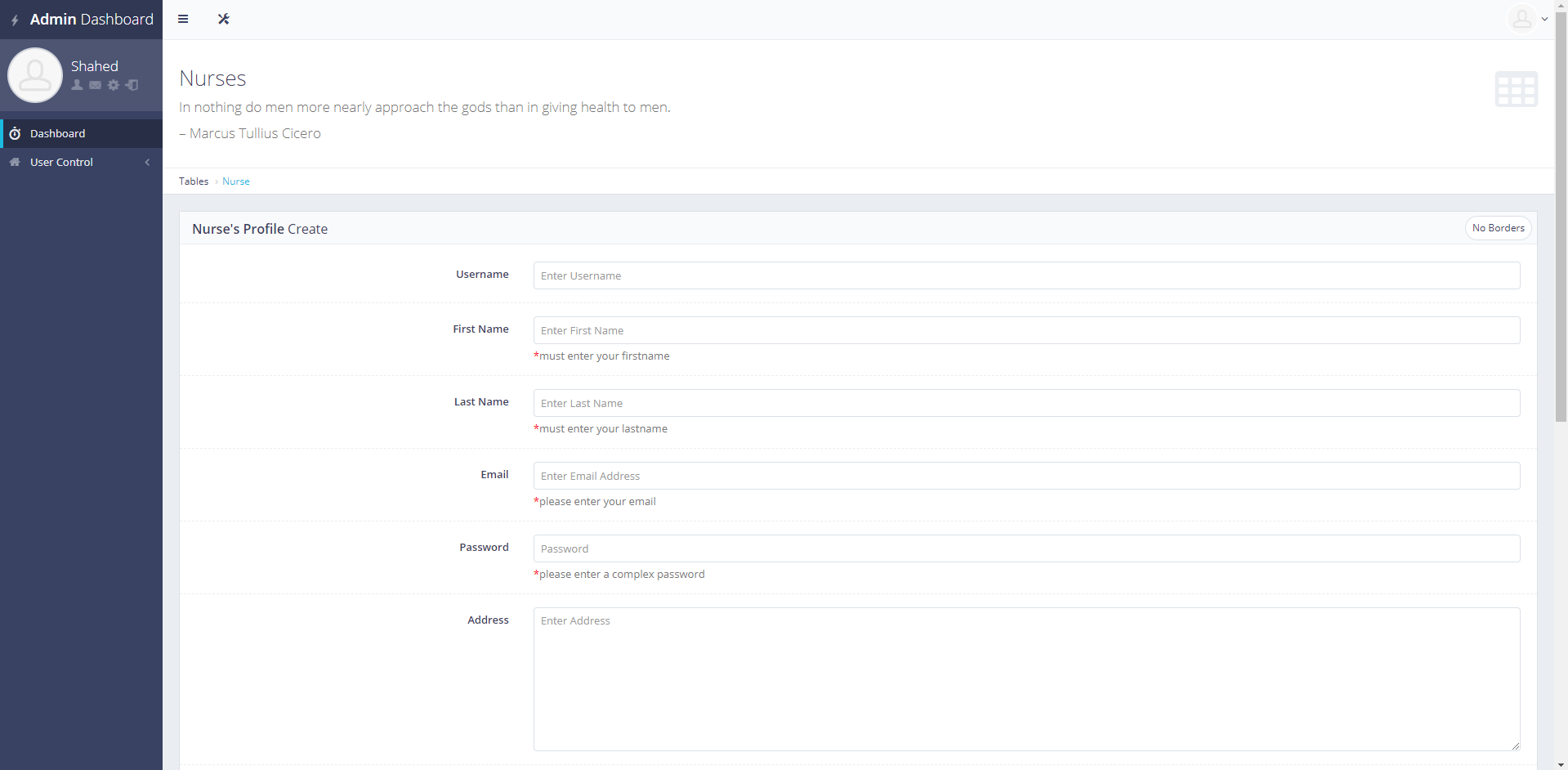


Figure 5.2.6

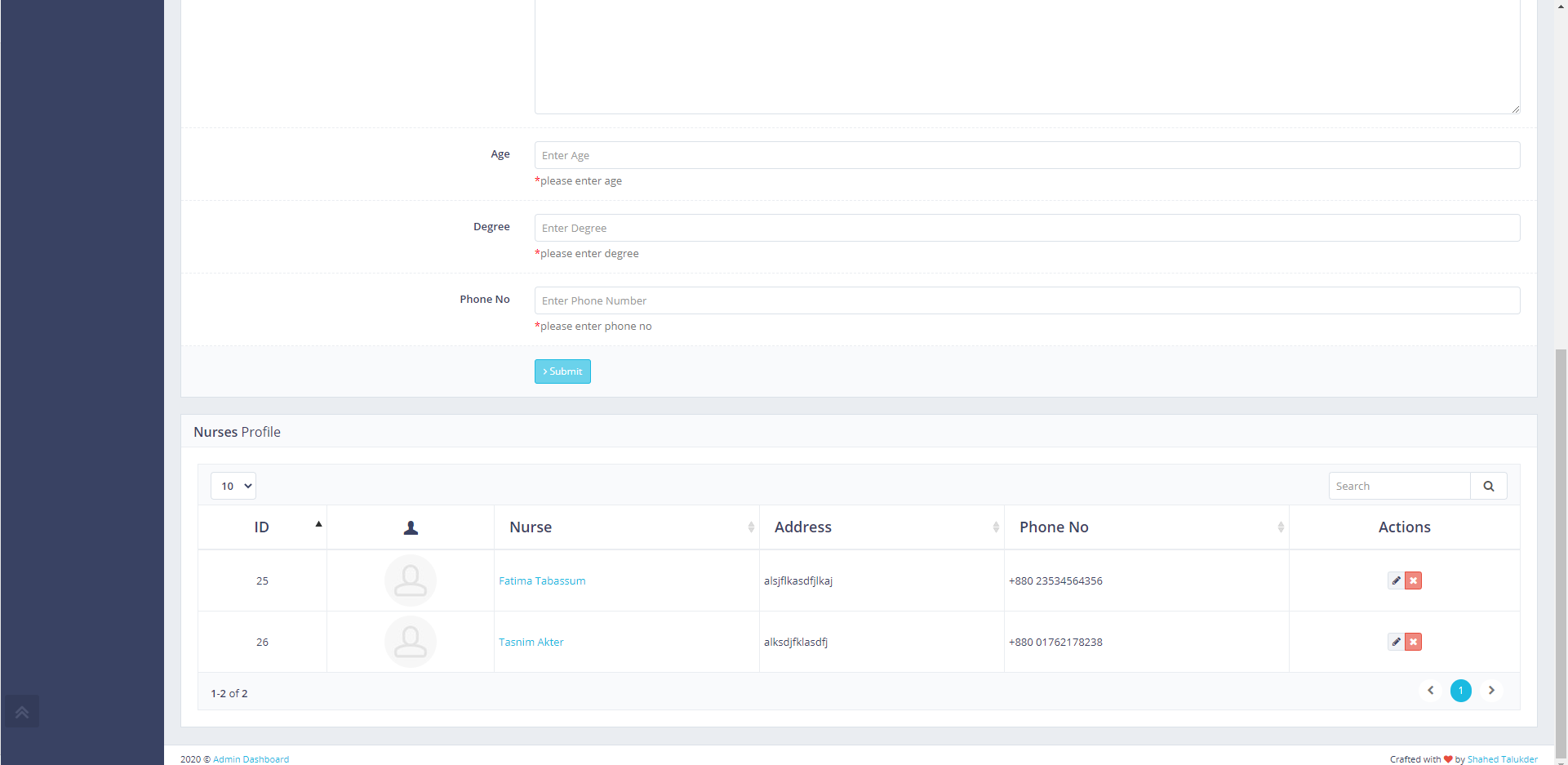


Figure 5.2.7

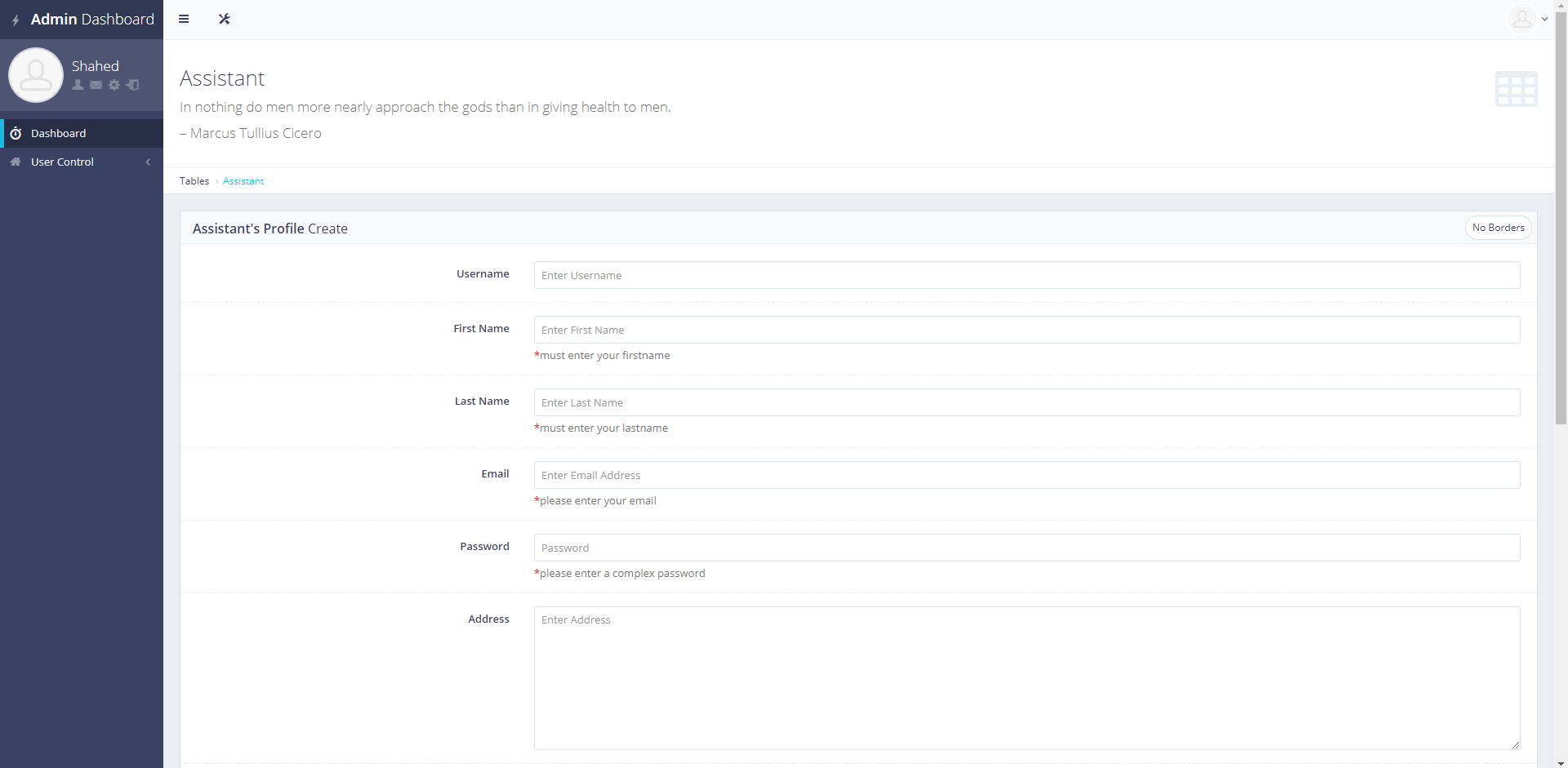


Figure 5.2.8

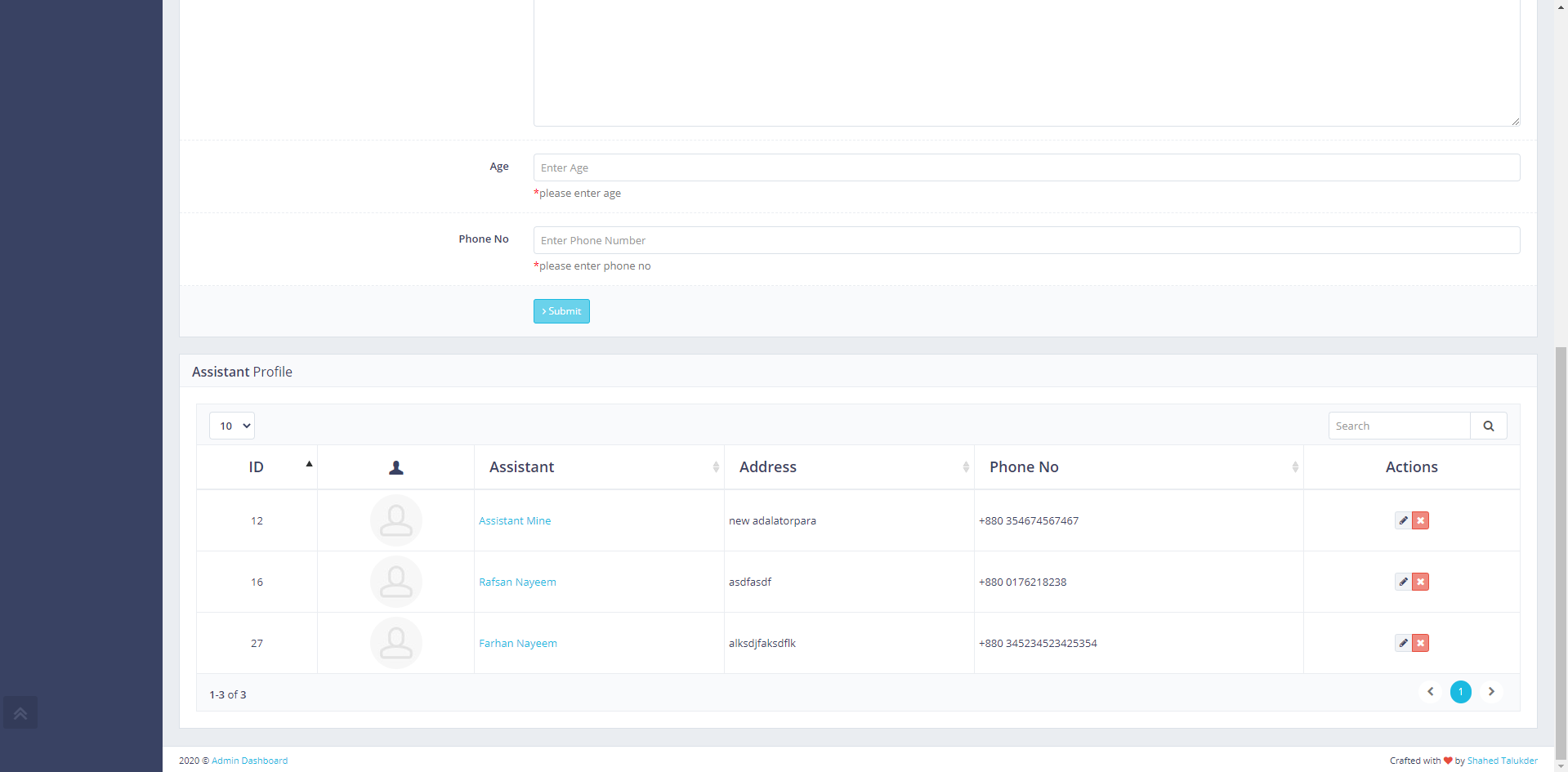


Figure 5.2.9

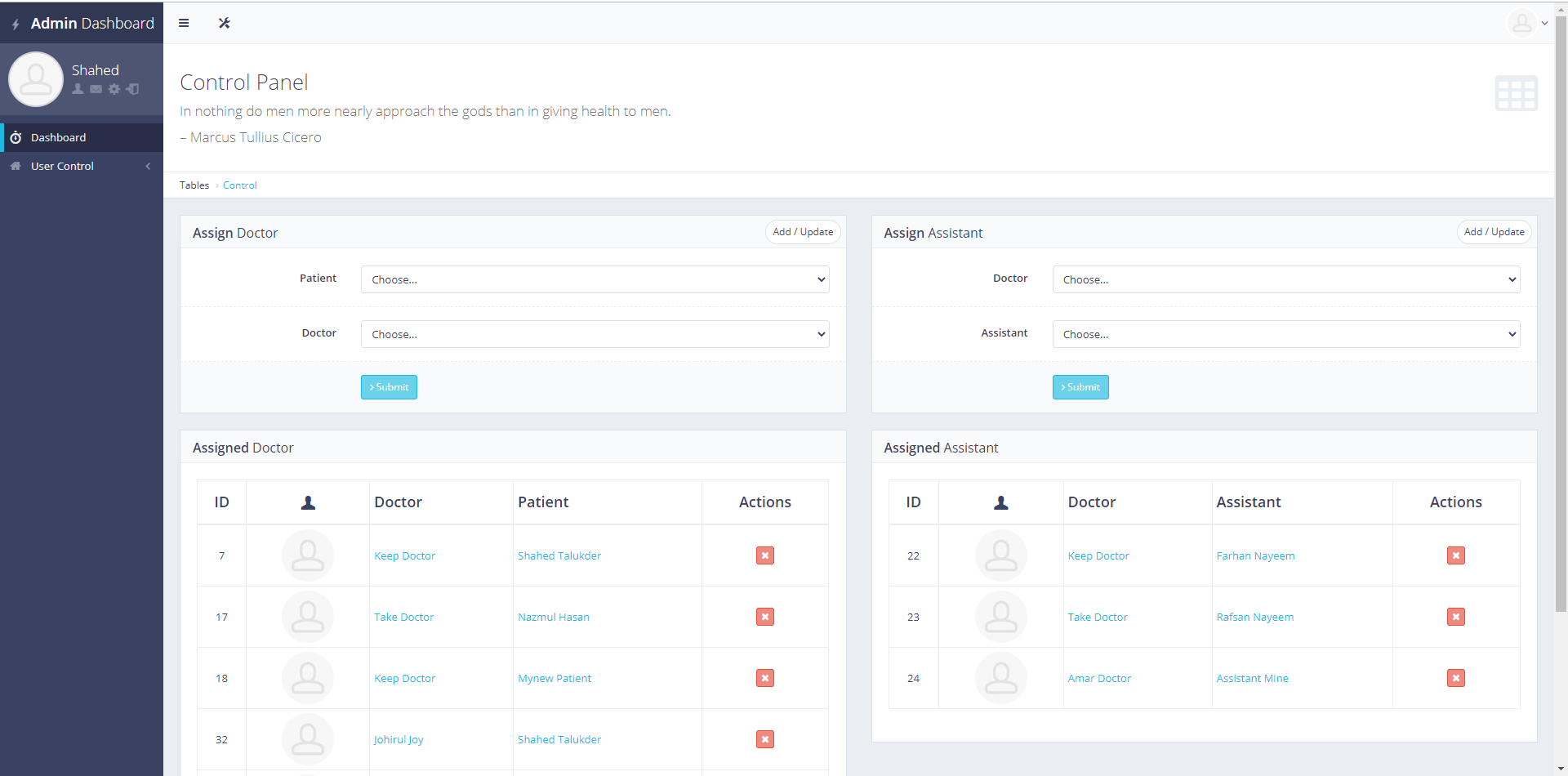


Figure 5.2.10

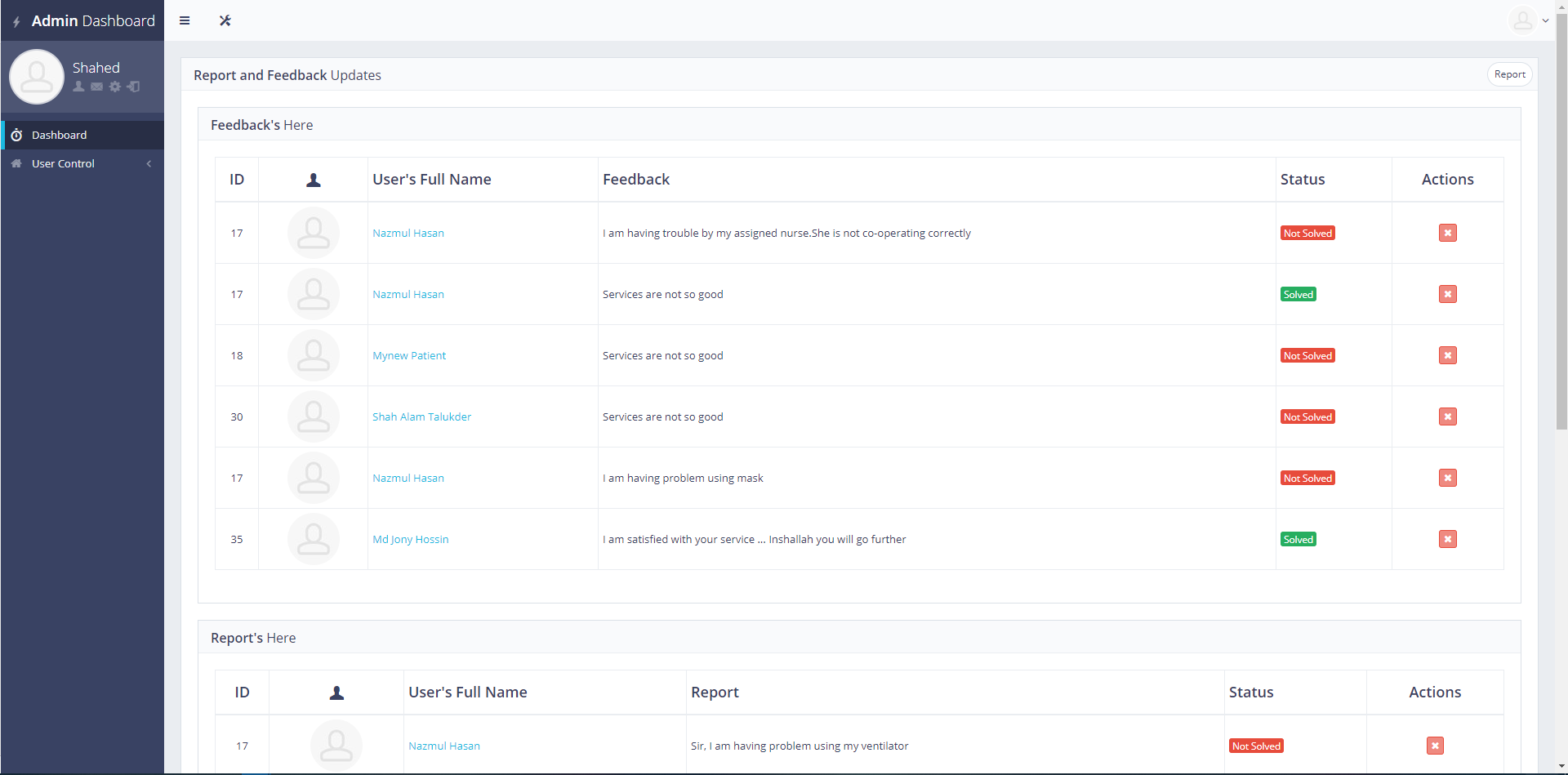


Figure 5.2.11

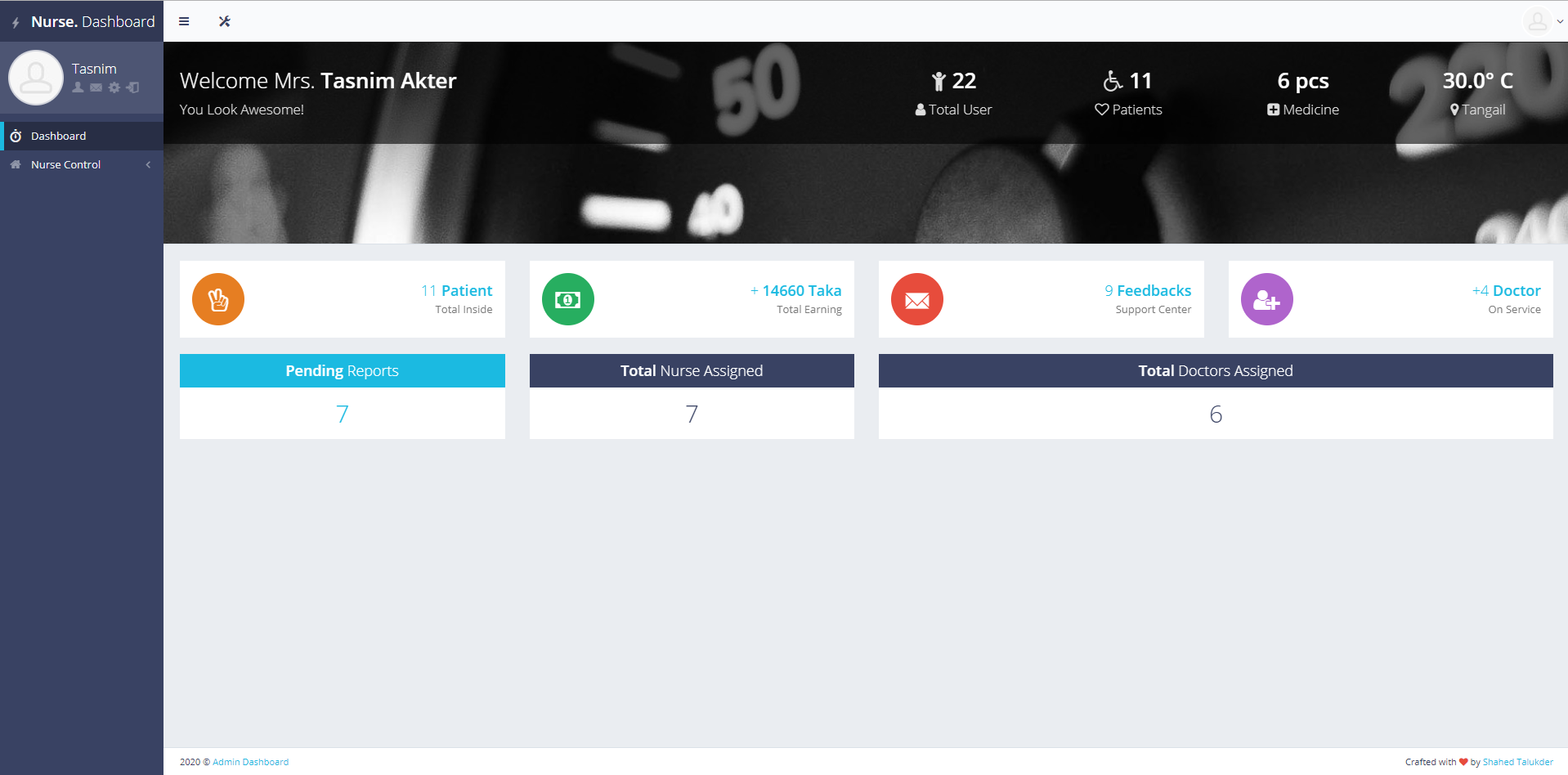


Figure 5.2.12

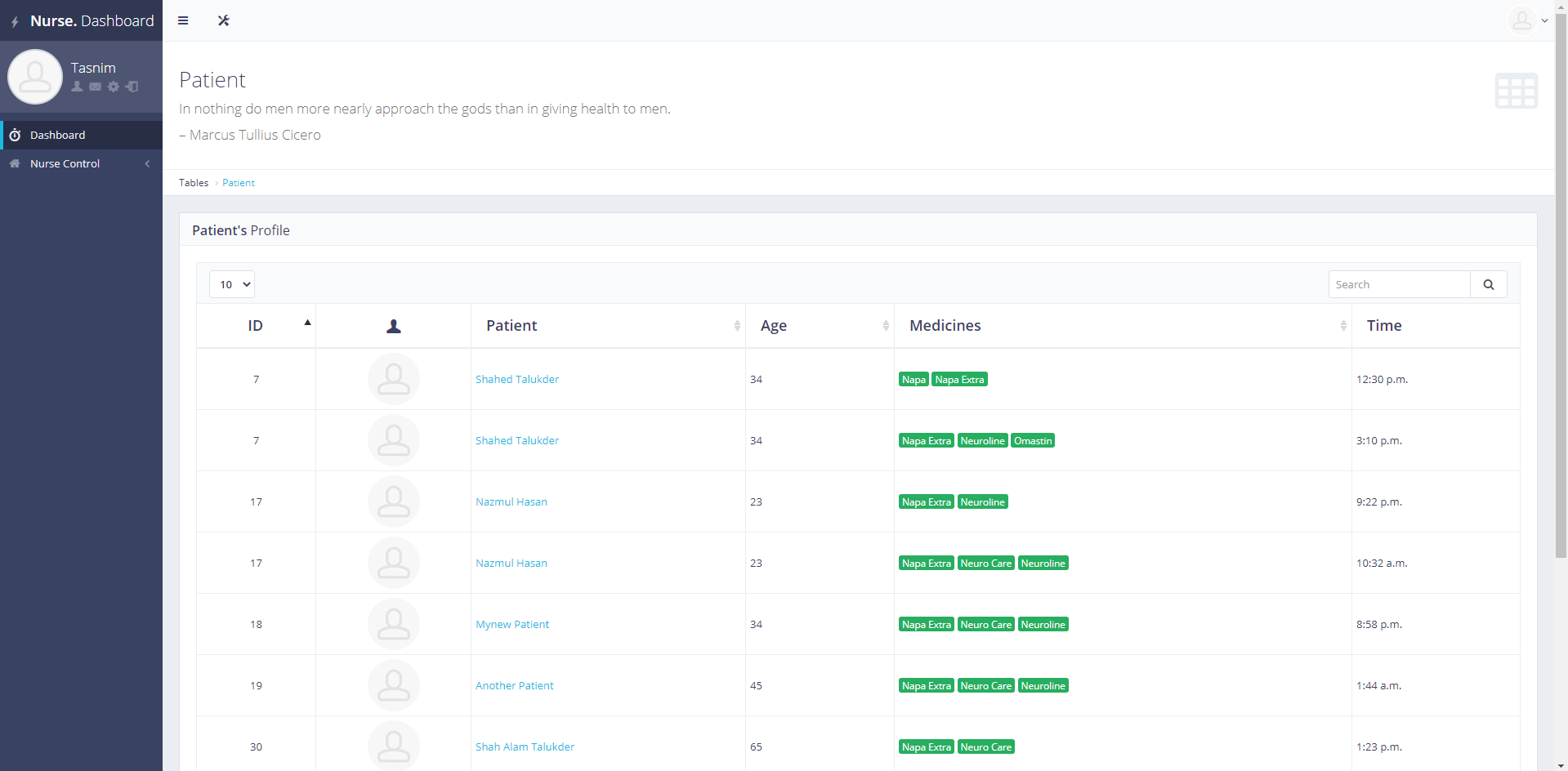


Figure 5.2.13

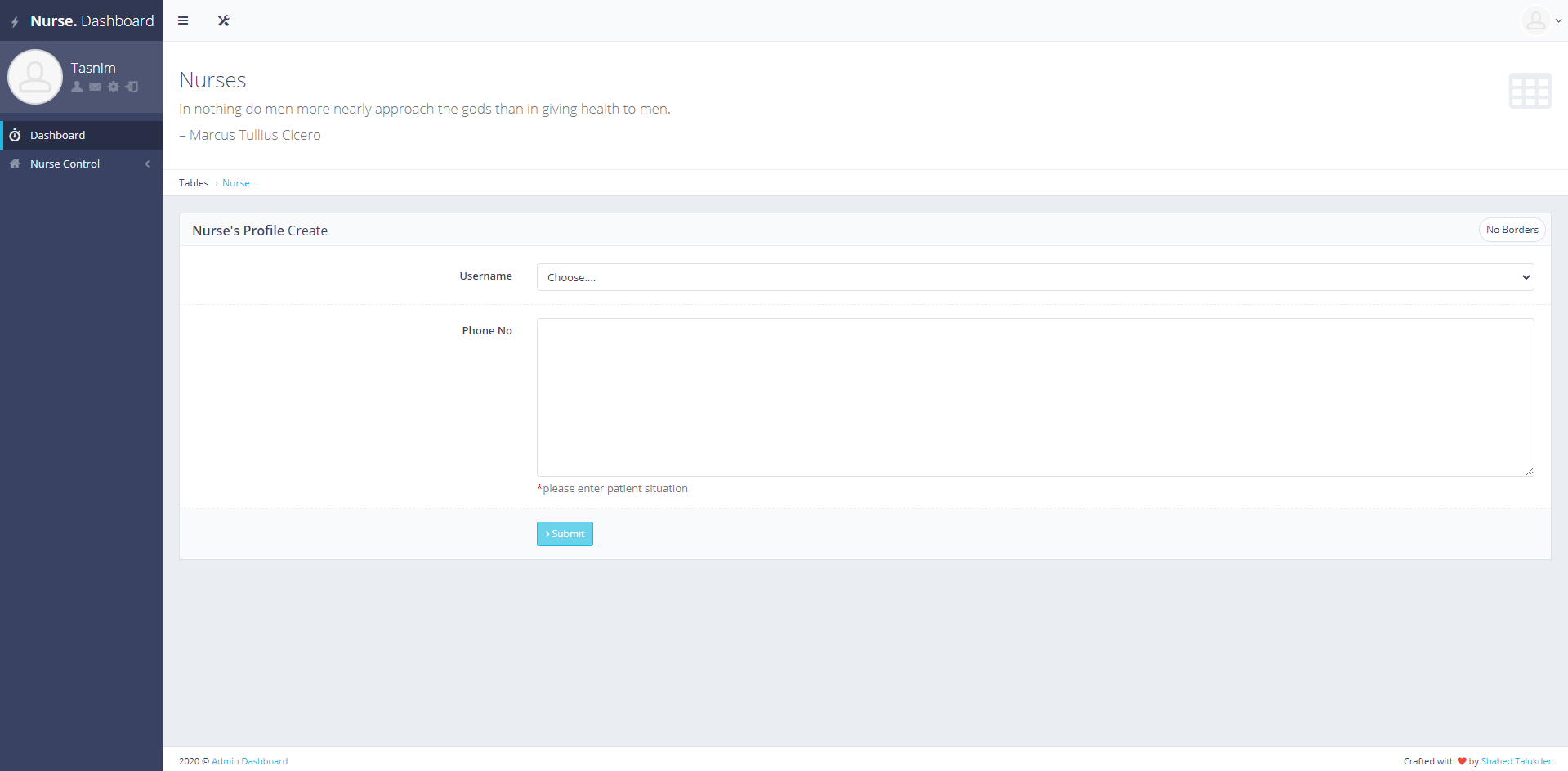


Figure 5.2.14

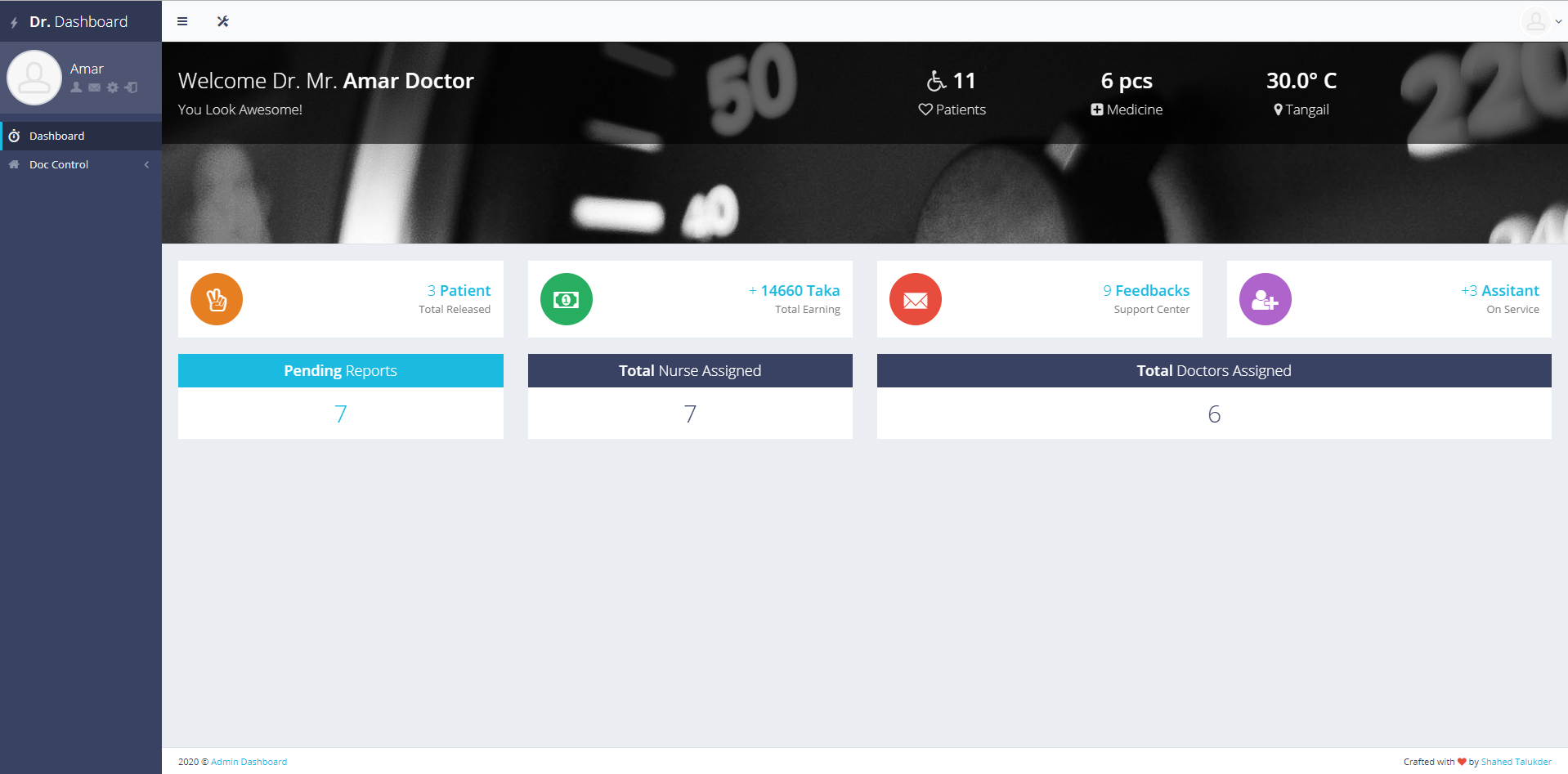
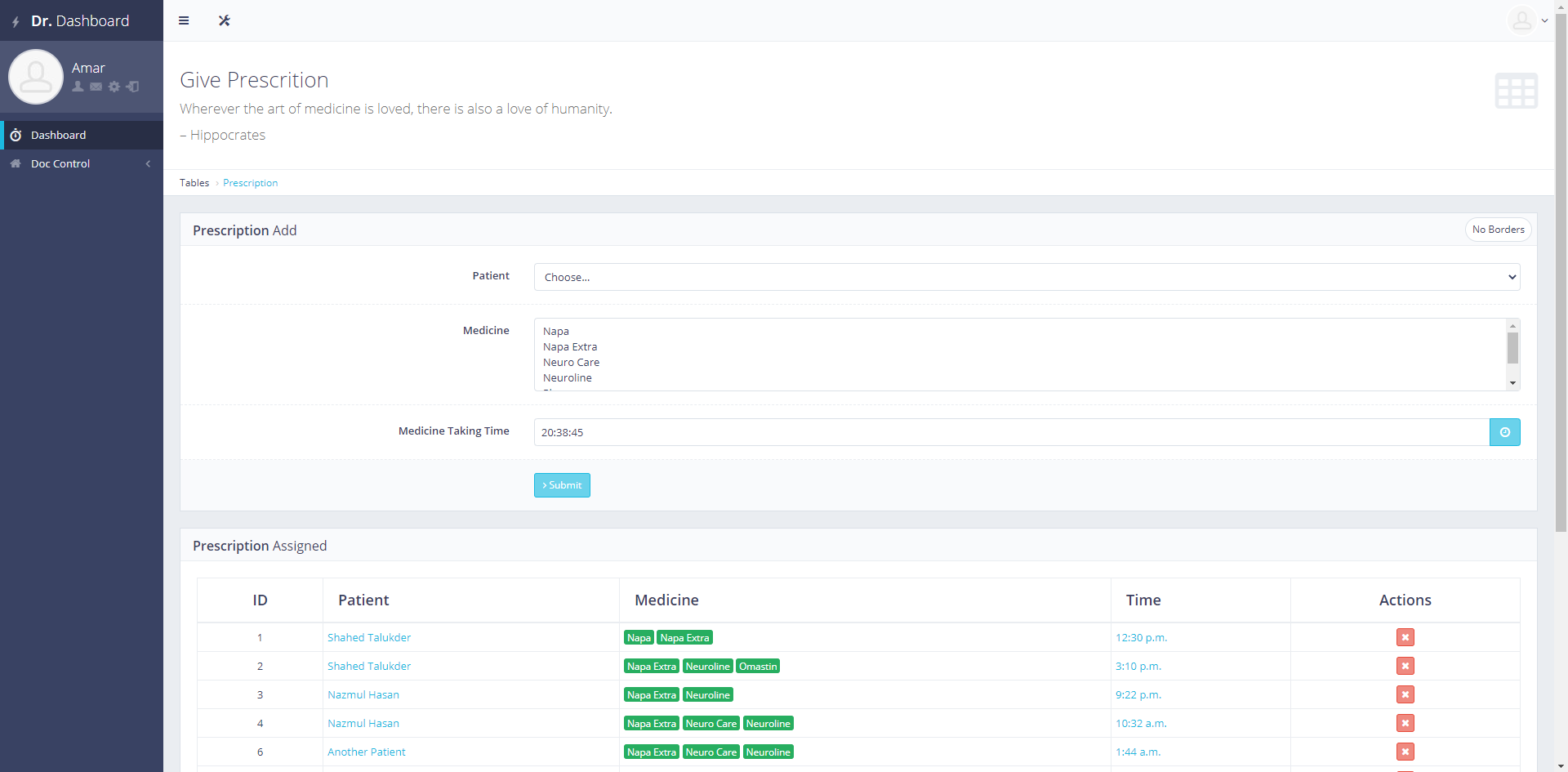


Figure 5.2.15

Figure 5.2.16

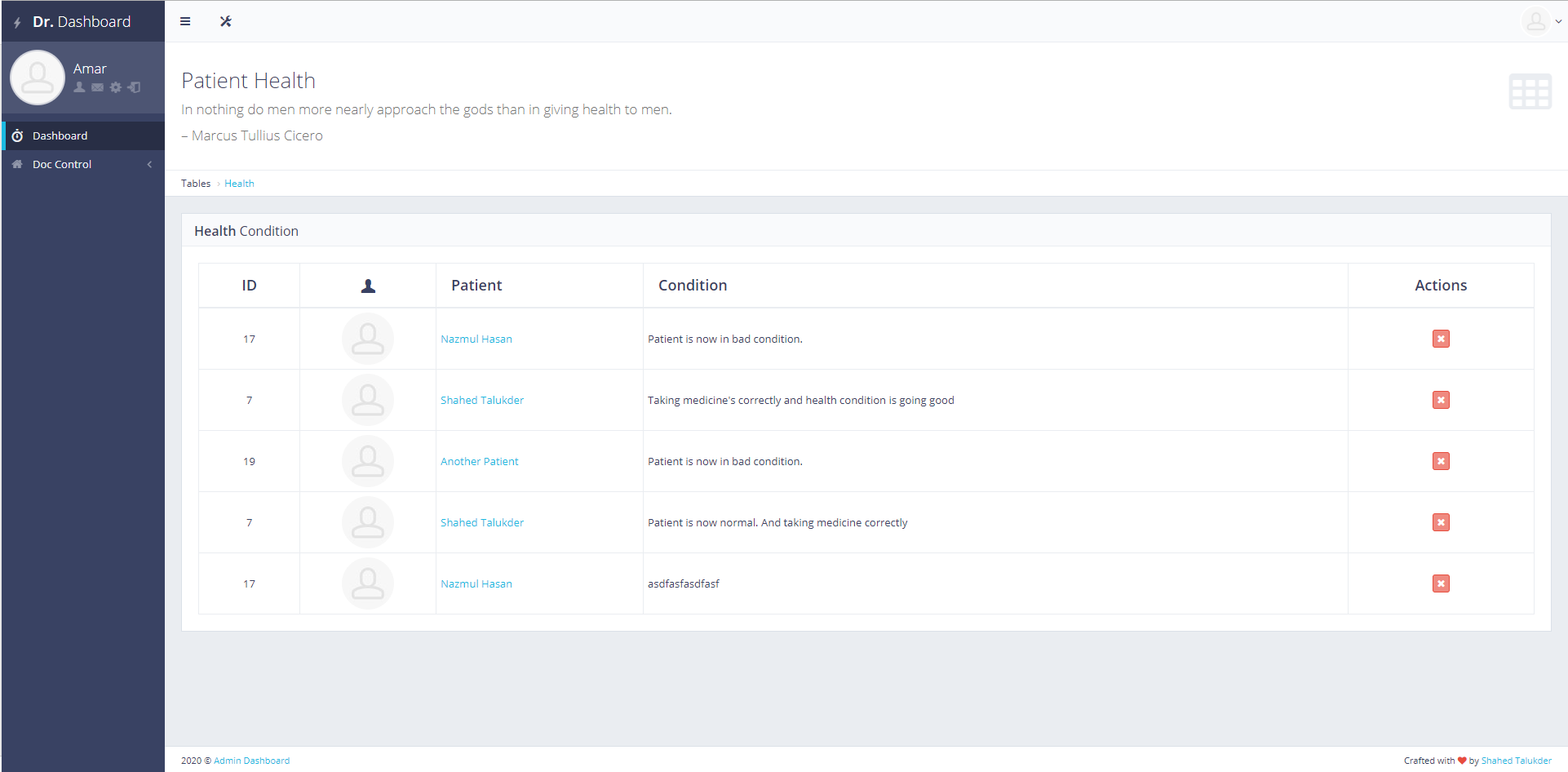


Figure 5.2.17

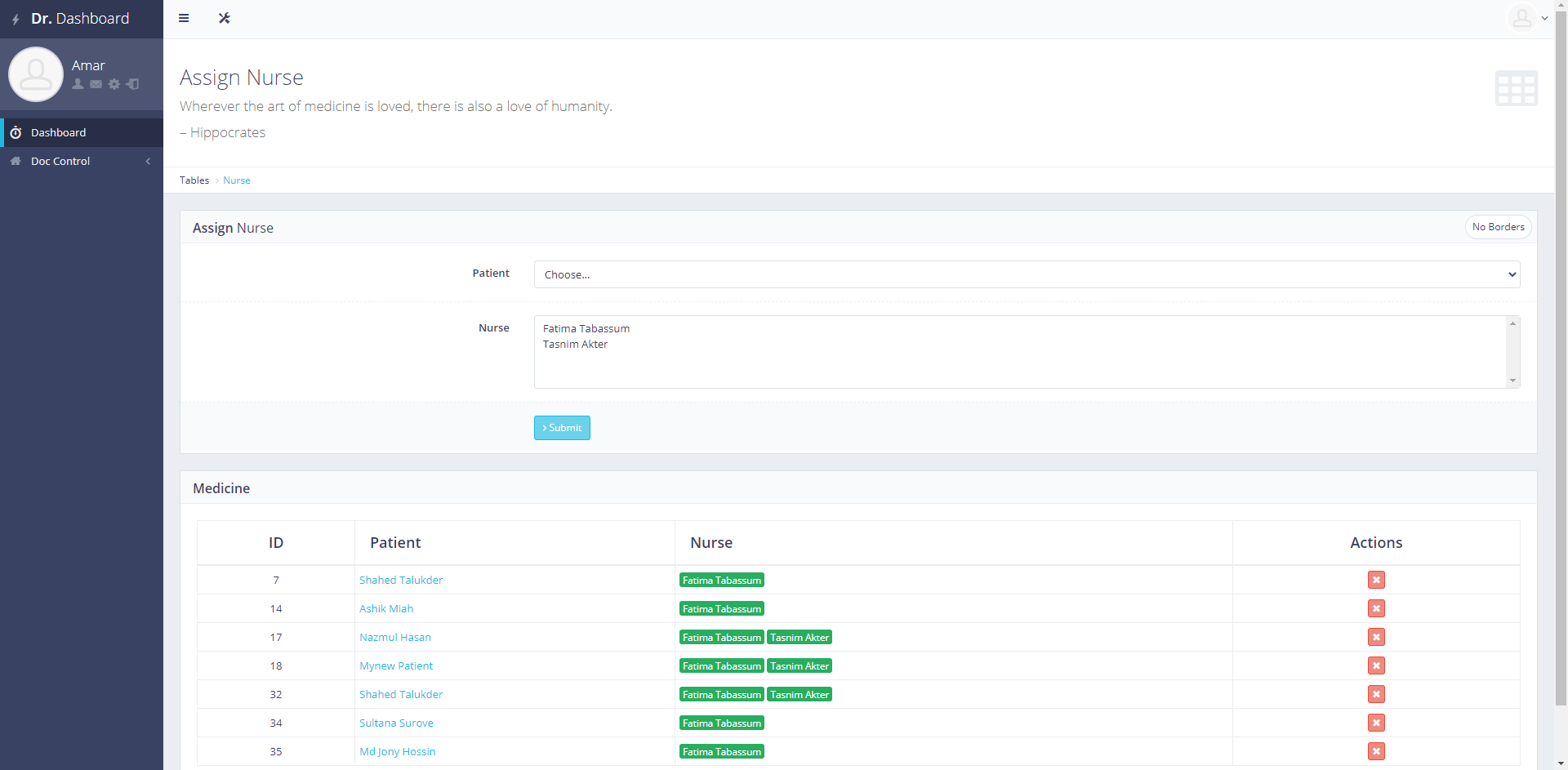


Figure 5.2.18

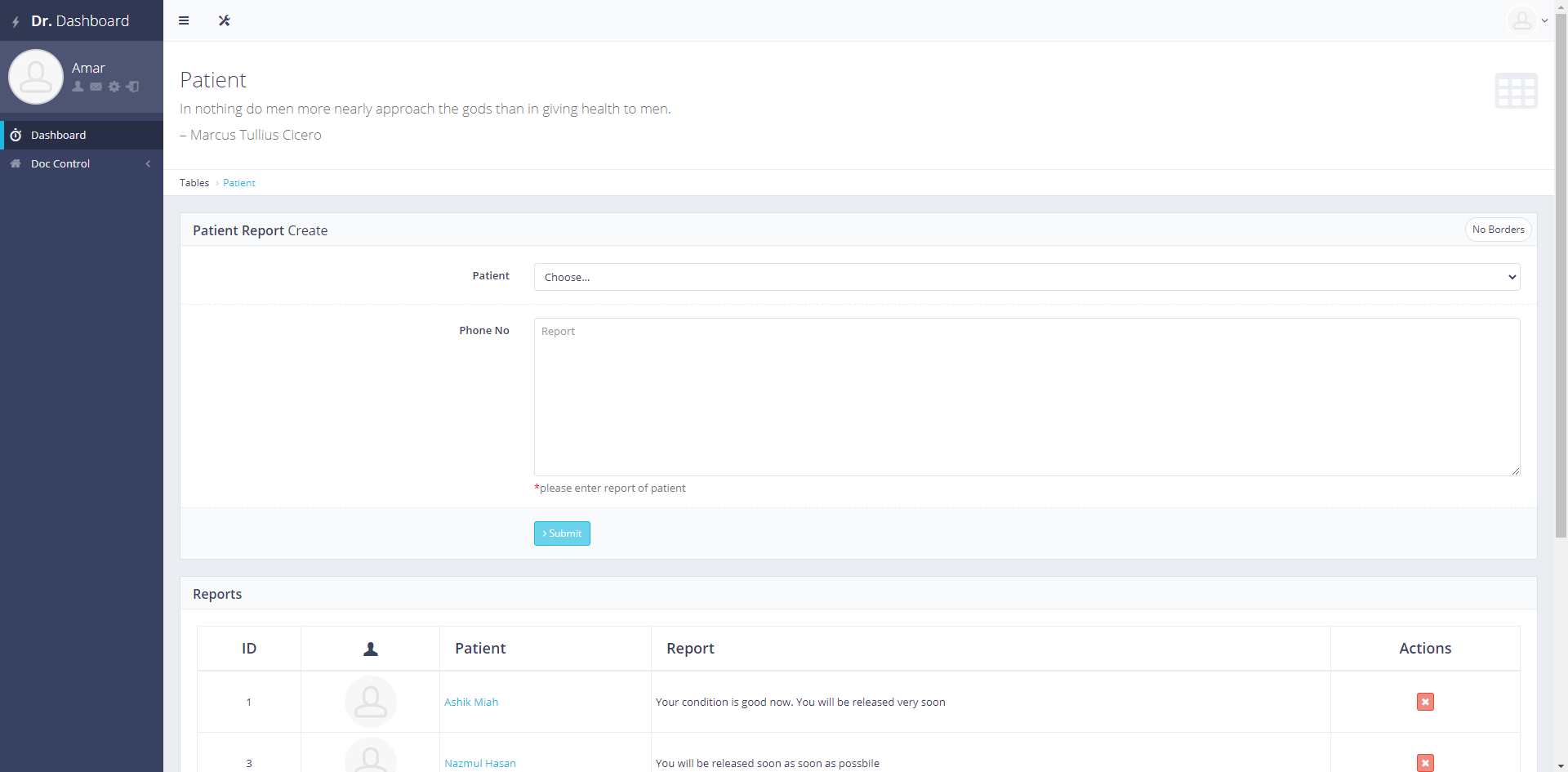


Figure 5.2.19

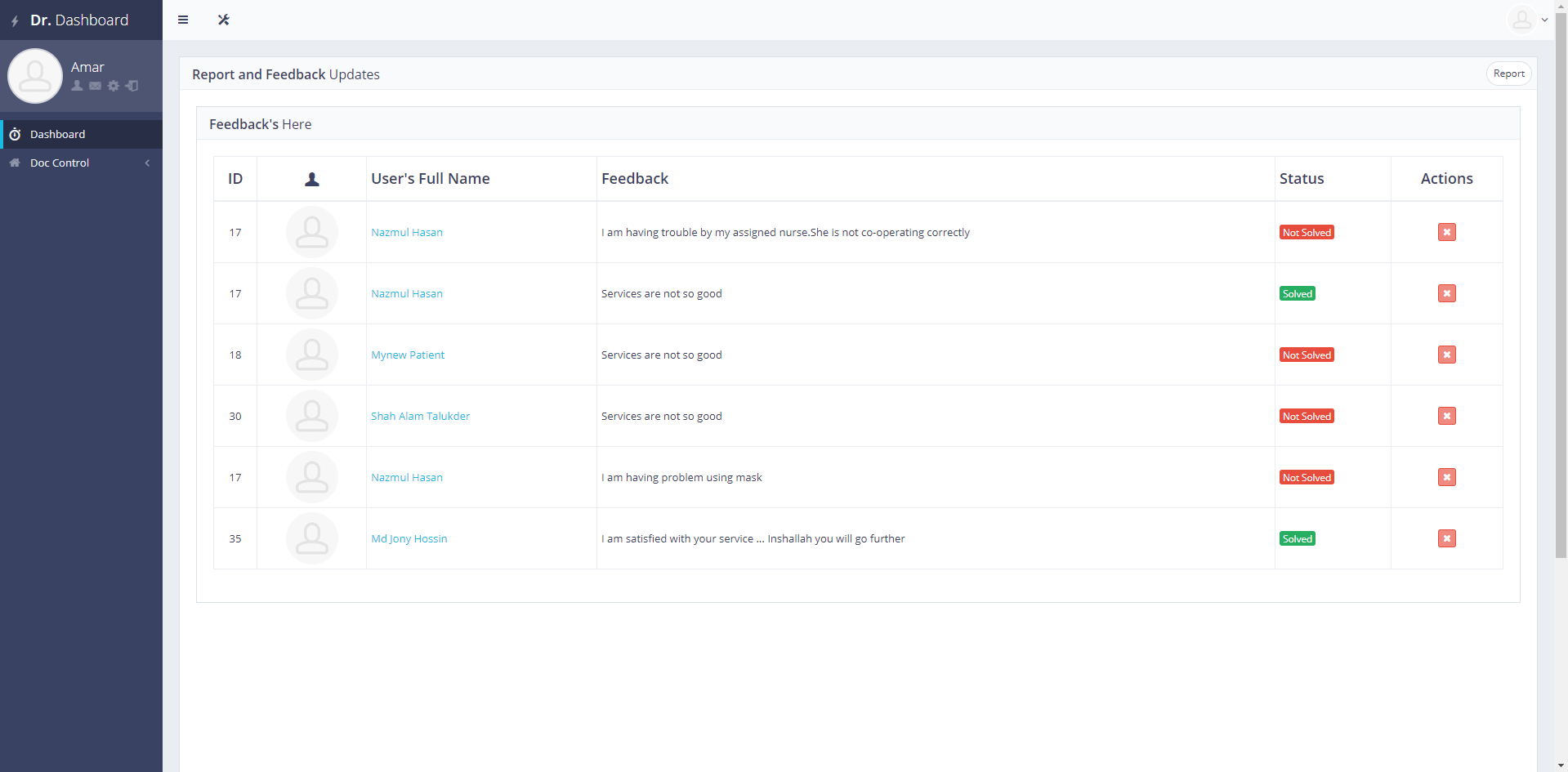


Figure 5.2.20

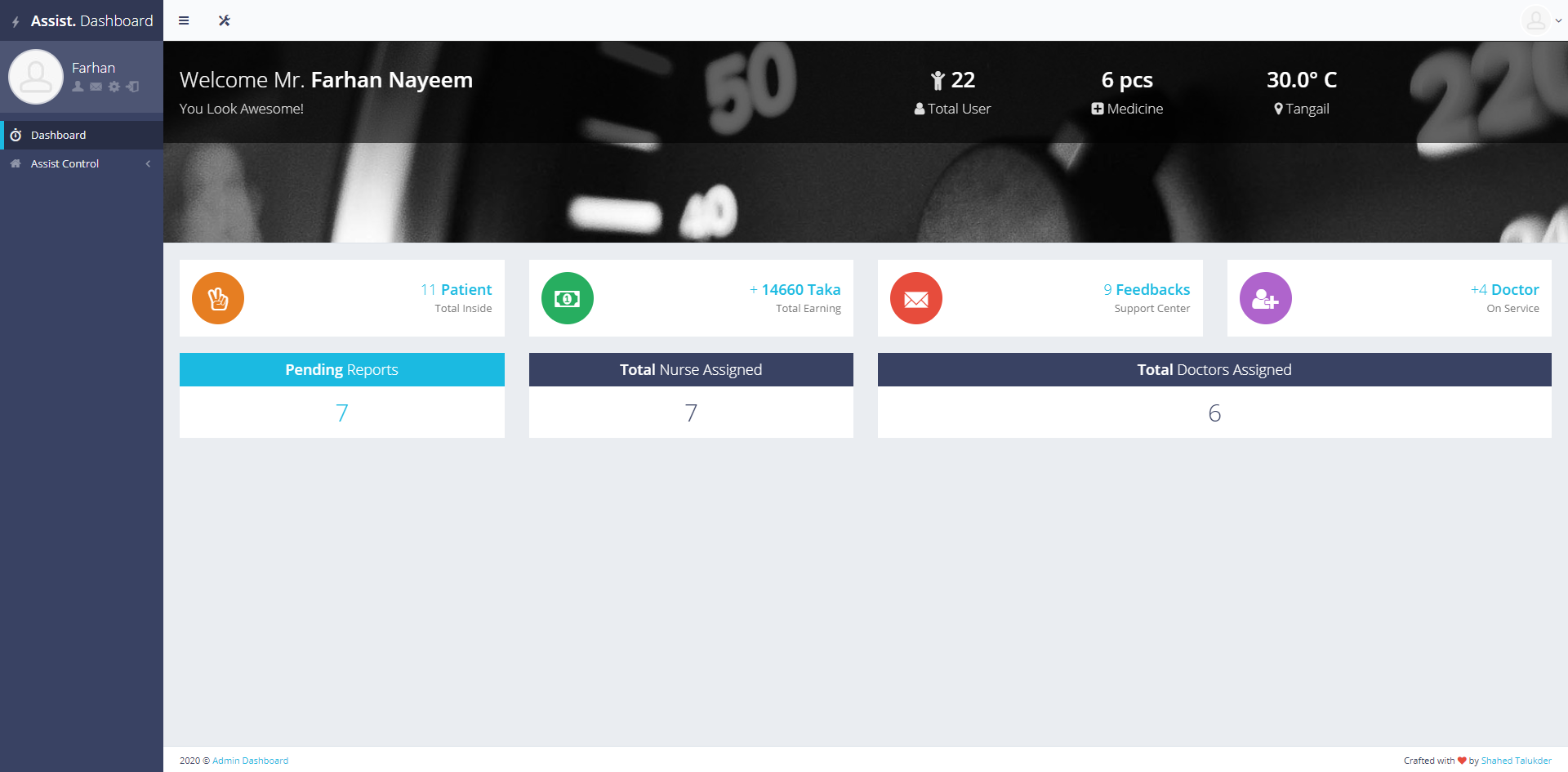


Figure 5.2.21

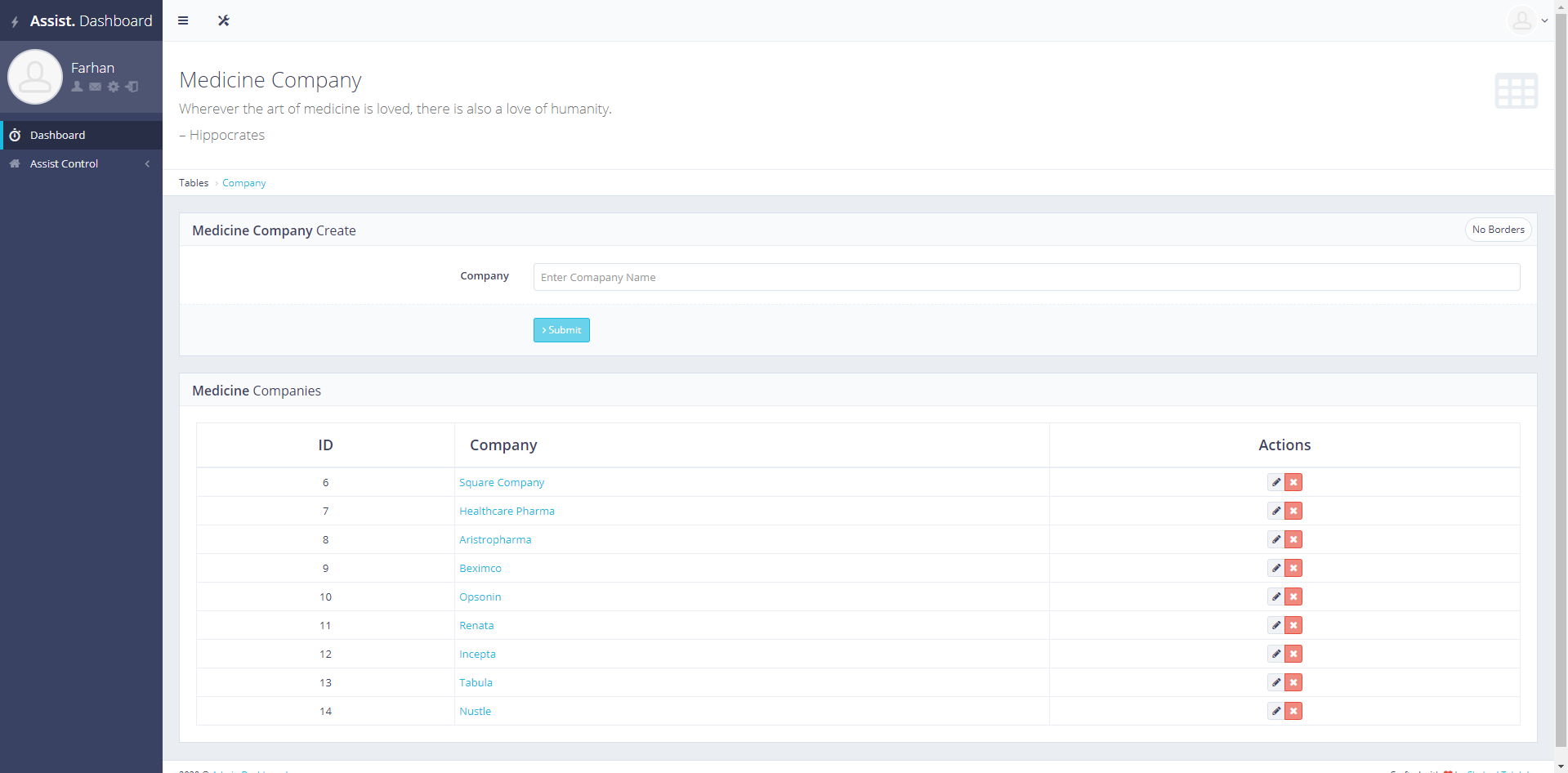


Figure 5.2.22

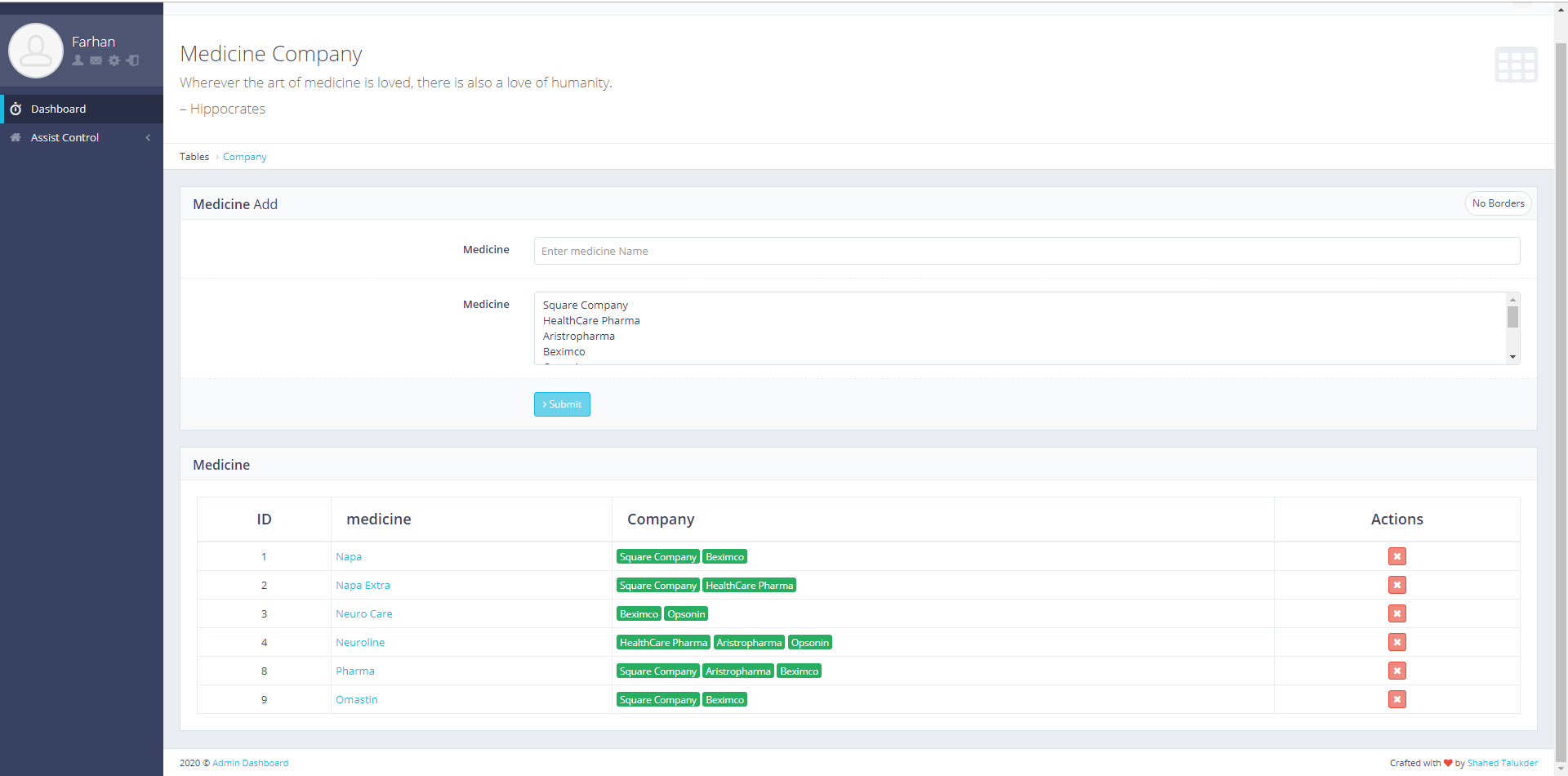


Figure 5.2.23

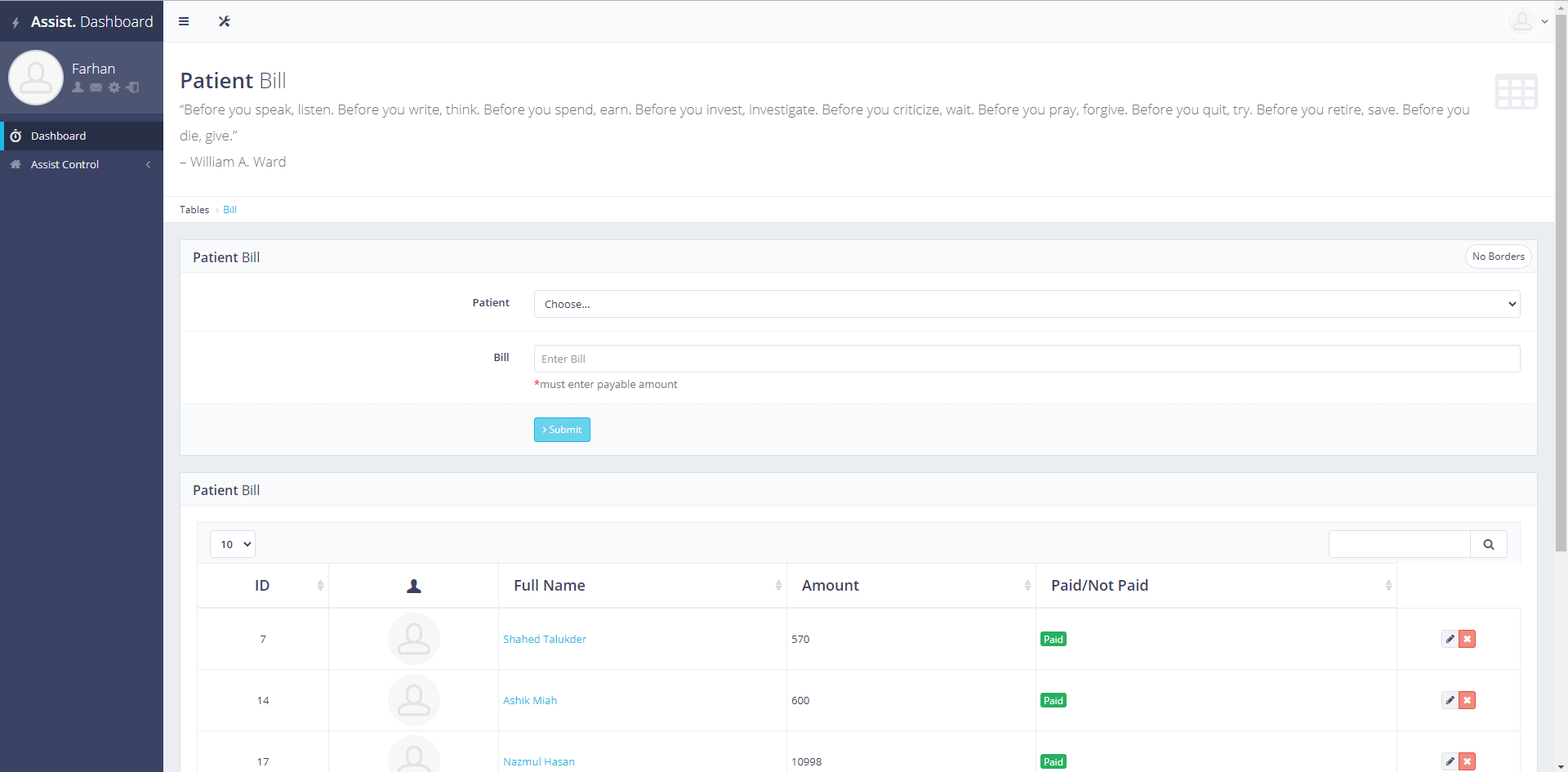


Figure 5.2.24

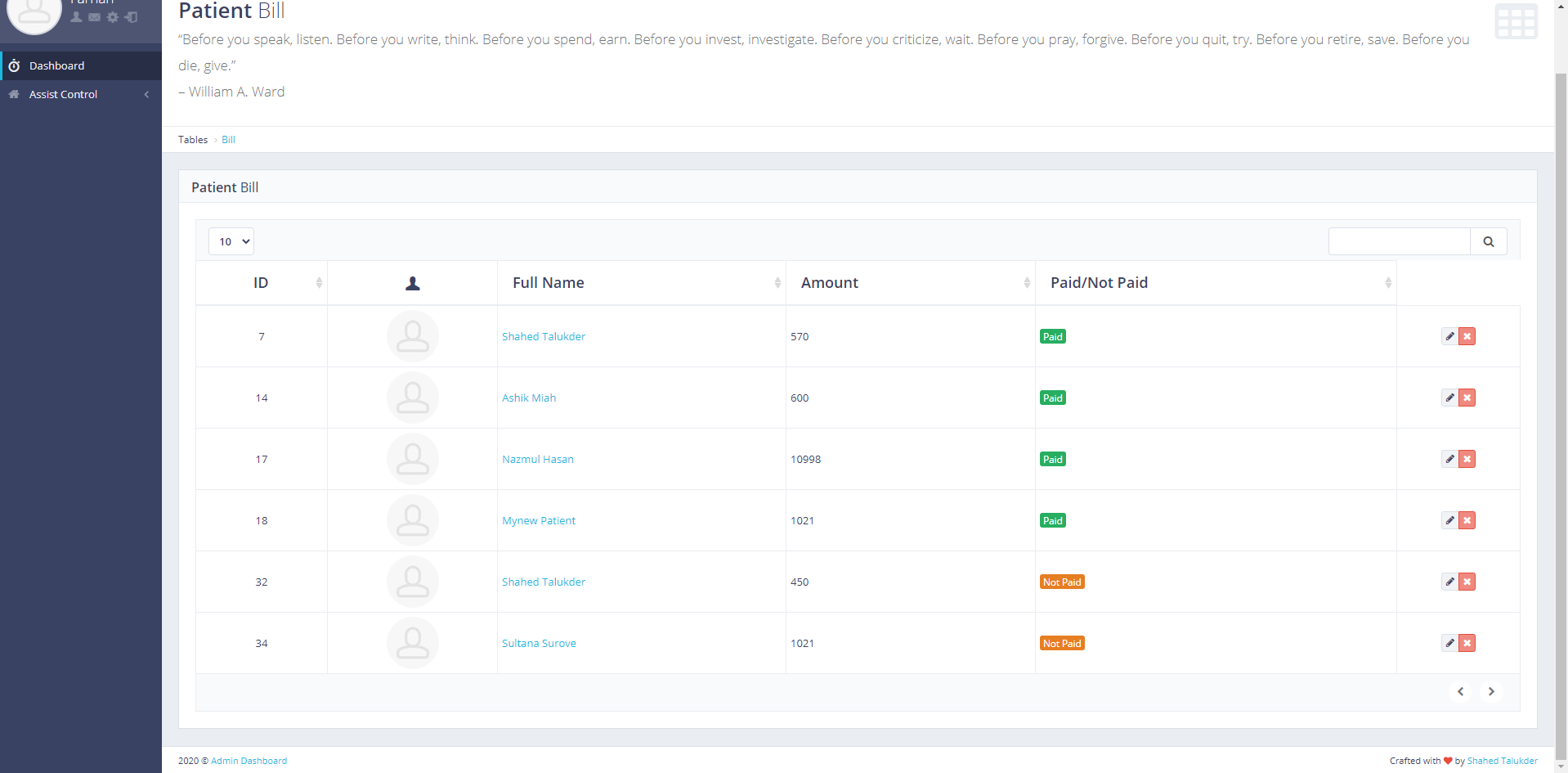


Figure 5.2.25

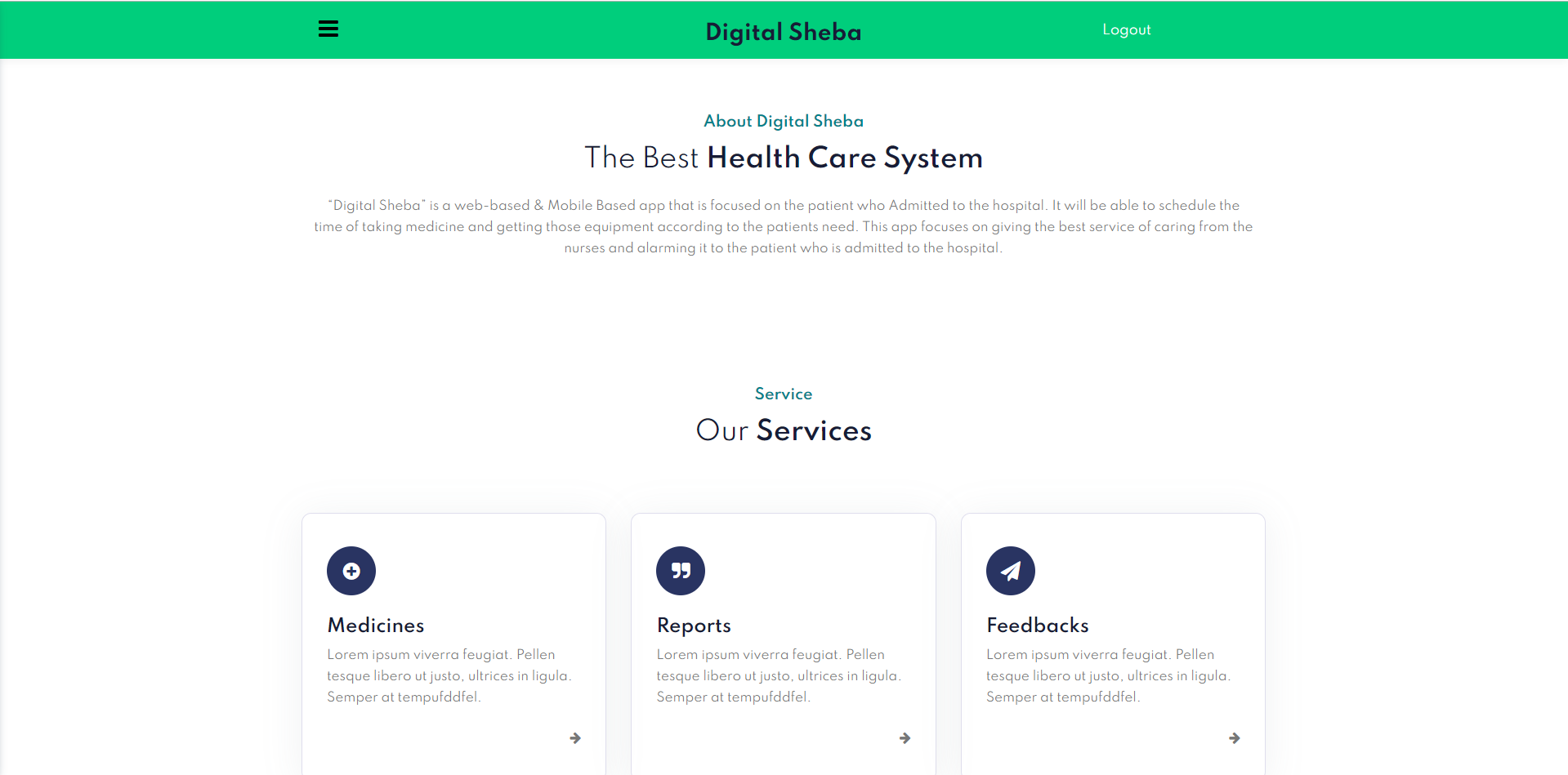


Figure 5.2.26



Figure 5.2.27

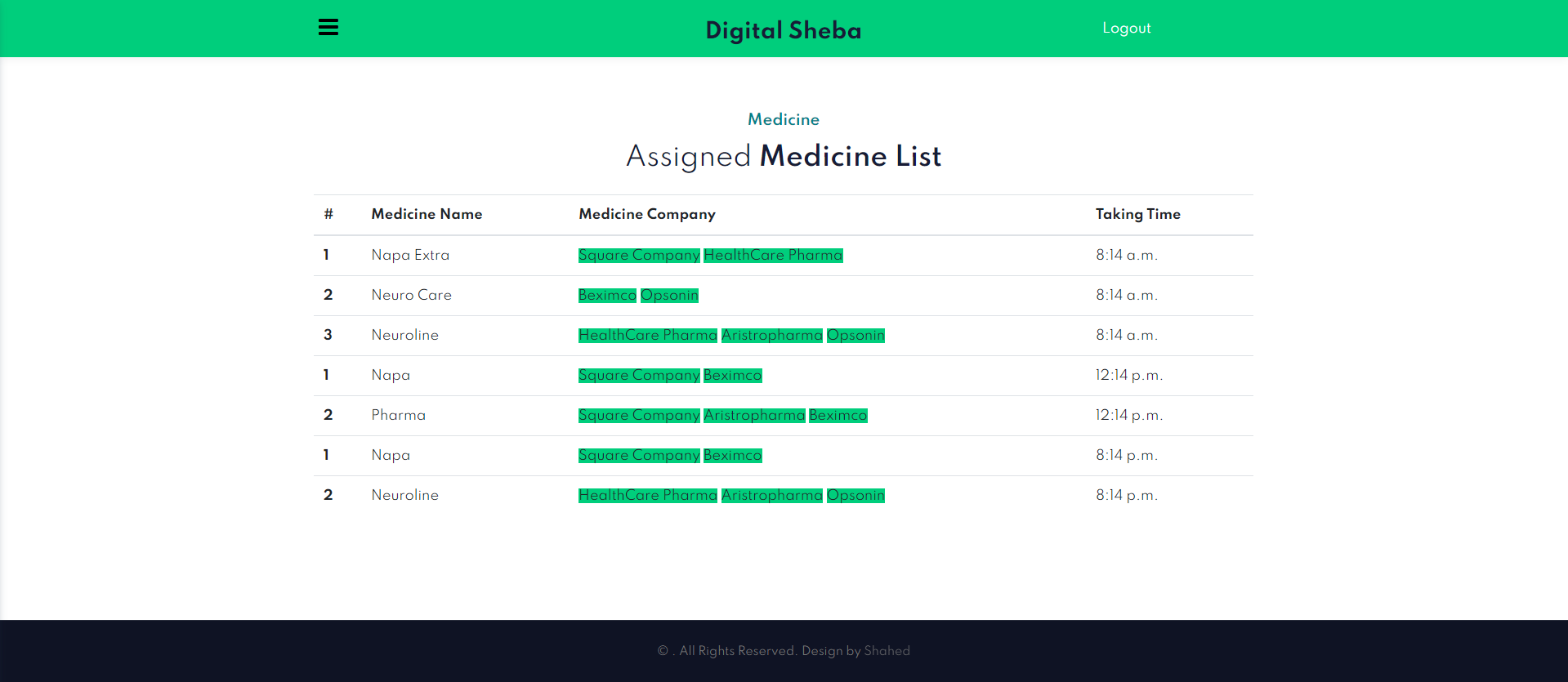


Figure 5.2.28

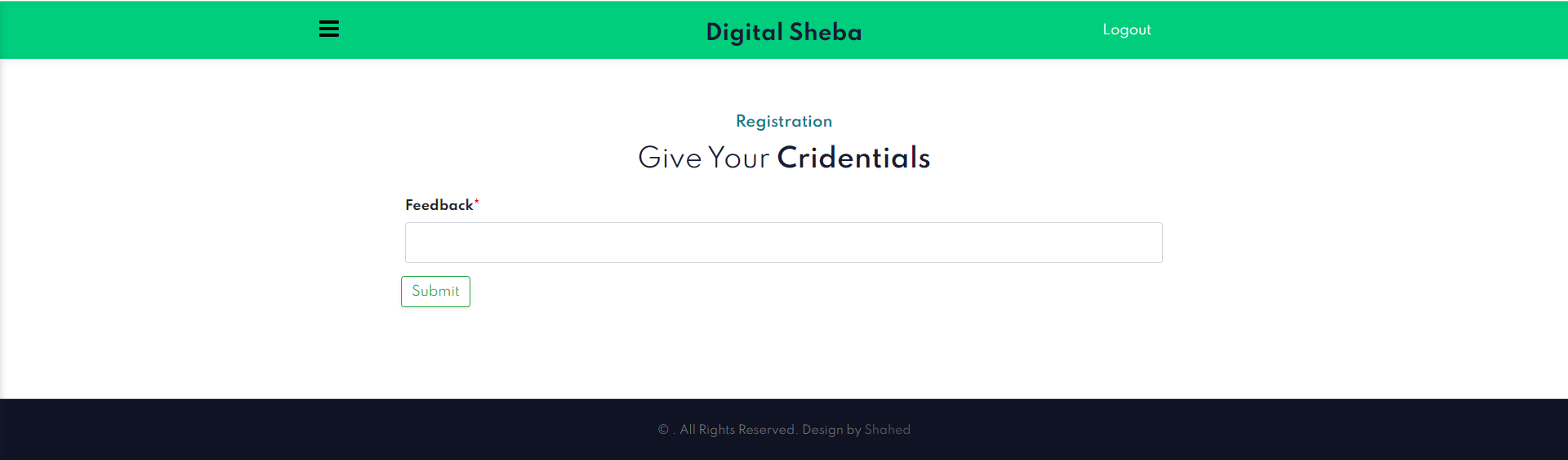


Figure 5.2.29

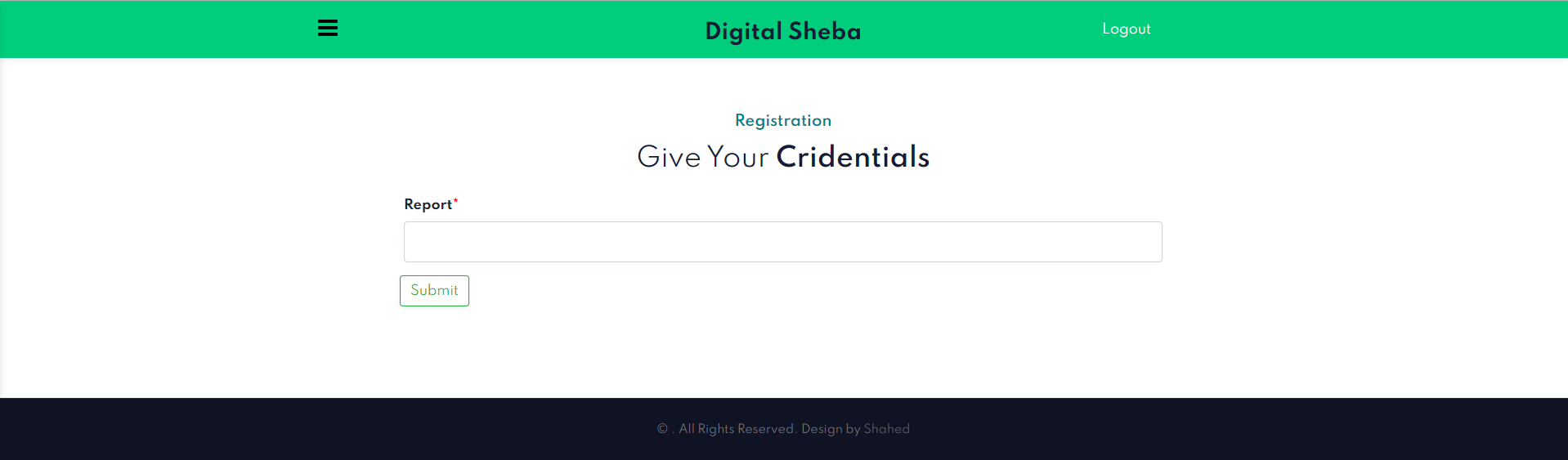


Figure 5.2.30

**5.3 Implementation of Interactions**

With the system from the very beginning the process of interactions are well planned. At the beginning, when the project is more conceptual, we do some workings. Ethnographic research, Surveying, focus group administration, Contextual Interview, Mental Modeling, Flow charts, Mood boards, Card sorting, Competitive analysis, Contextual Inquiry was done.

While the project is n development phase: Wireframing, Heuristic analysis. Expert evaluation. Personas, Scenario, Prototypes of two stage: Alpha & Beta version, System mapping, Experience mapping & User testing / usability testing has been also done.

When we have completed the Beta version & user acceptance testing, we do some minor user interaction changes, patch some wireframing & fine-tuning.

Thus, the system met proper & valid visual design, user experience & human computer interaction, usability and accessibility.

**5.4 Testing Implementation**

Testing is a simple term in checking web application for potential bugs before its made live or before code is moved into the production environment.

During this stage; our developed system issues such as that of web application security, the functioning of the site, its access to handicapped as well as regular users and its ability to handle traffic is checked. We tested the functions security, interface, database & other requirements as below:

* Functionality Testing:

We tested outgoing links, internal links, anchor links & others to check the system functionality. We also check Forms, Cookies, Front-end functions.

* Usability Testing:

Uses of our functions, features, contents & other in position is checked in this stage. We did test of navigation & site pages redirecting.

* Performance Testing:

Website loading response time is short, cached memory & cookies is reusable. The overall performance is satisfactory for a user.

* Security Testing:

Data passes through user and server end point is well secured. We are using encryption-based data passing for ensuring higher security for the system. So, we divided the system backend in two parts. Also, the admin & super admin have secure login options.

**5.5 Test Results and Reports**

We have completed our project & the project meets all the requirements. The final version of Course Tracking Management System have no bugs & interaction problems. The testing of the web application is successful & have found no error. The results of the CTMS is given bellow.

* **Completion of Percentage**: We have completed our project 100%, we have met all the functional requirements that we discussed in the requirement section.
* **Percentage of Accuracy:** Our project is working 100% accurate. It fulfills all the functional & non-functional requirements as we mentioned in the previous chapter earlier.
* **Percentage of Correctness:** As we have tested al the requirements and make their test cases mentioned and clear all the mistakes so now our project is 100% correct.

**CHAPTER 6: CONCLUSION AND FUTURE SCOPE**

**6.1 Discussion and Conclusion**

The project **“Digital Sheba”** is for computerizing the working in a hospital. It is a great improvement over the manual system. The computerization of the system has speed up the process. In the current system. The front office managing is very slow. The hospital managing system was thoroughly checked and tested with dummy data and thus is found to be very reliable. The software takes care of all the requirements of an average hospital and is capable to provide easy and effective storage of information related to patients that come up to the hospital.

It generates test reports and also provides the facility for searching the details of the patient. It also provides billing facility on the basis of patient’s status whether it is indoor or outdoor patient. The system also provides the facility of backup as per the requirement.

**6.2 Scope of Future Development**

Though our requirements meet the full thought of our objectives & motivation there are some scope to improve the system & easier for the user. The further scope of developments are our future developments for this project.

Scope for further developments:

* Mobile Application: For extending portability, flexibility and more time saving the development of an application for mobile phones is matter of time. Everyone wants a system in their hands for easy usability. This Mobile application will met the desire.
* Adding Feature: The proposed system can be enhance by including more facilities like pharmacy system for the stock details of medicines in the pharmacy.