A MAJOR PROJECT REPORT ON

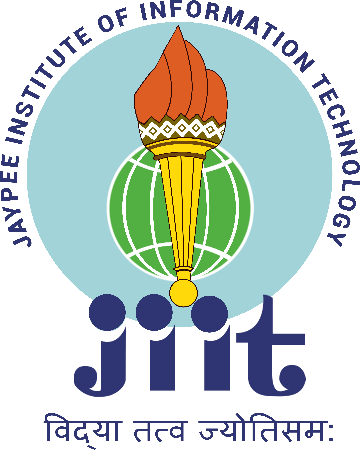
**E-DIAGNOSIS: ONE STOP WELLNESS**

SUBMITTED IN PARTIAL FULFILLMENT FOR THE AWARD OF DEGREE OF

**BACHELOR OF TECHNOLOGY**

**IN**

### ELECTRONICS AND COMMUNICATION ENGINEERING



**SUBMITTED BY: UNDER THE GUIDANCE OF:**

CHAYAN GANDHI (9915102070)

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA (U.P.) MAY 2019**

**CERTIFICATE**

This is to certify that the major project report entitled, **E-DIAGNOSIS: ONE STOP WELLNESS** submitted by **CHAYAN GANDHI (9915102070)** in partial fulfillment of the requirements for the award of Bachelor of Technology Degree in **Electronics and Communication Engineering** of the Jaypee Institute of Information Technology, Noida is an authentic work carried out by them under our supervision and guidance. The matter embodied in this report is original and has not been submitted for the award of any other degree.

**Signature of Supervisor:**

**Dr. Jitendra Mohan**

**Department of Electronics & Communication**

**JIIT, Sec-128, Noida-201304**

**Dated:**

# DECLARATION

I hereby declare that this written submission represents my own ideas in our own words and where others' ideas or words have been, included, have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission.

Place:

Date:

Name: Chayan Gandhi

Enrollment: 9915102070

Name: Karan Gupta

Enrollment: 9915102072

Name: Vipul Singh

Enrollment: 9915102073

## ACKNOWLEDGMENT

I would like to express my special thanks of gratitude to Dr. Jitendra Mohan who gave us the golden opportunity to do this wonderful project on the topic **E-DIAGNOSIS: ONE STOP WELLNESS**, which also helped us in doing a lot of Research and I came to know about so many new things I are really thankful to them. I would also like to express my sincere gratitude to my lab technicians for their help during the course by providing their expert academic and practical guidance.

Chayan Gandhi (9915102070)

Karan Gupta (9915102072)

Kapil Gupta (9915102073)

TABLE OF CONTENTS

Certificate i

Declaration ii

Acknowledgement iii

List of Figures & Tables iv

Abstract v

CHAPTER 1. Introduction 1-4

1.1 Problem Statement 1

1.2 Need of System 2

1.3 Literature Survey 2-4

CHAPTER 2. System Architecture 5-11

2.1 Arduino NANO 5

2.2 Alcohol Sensor 7

2.3 DC Motor 8

2.4 LCD Display 9

2.5. Arduino IDE 10

CHAPTER 3. Experimental Details 12-13

CHAPTER 4. Results 14-15

4.1. Test Case: Initial Stage with No Alcohol 14

4.2. Test Case: Initial Stage with Alcohol below limit 15

4.3. Test Case: Initial Stage with alcohol above limit 15

4.4. Test Case: Running engine stopped 15

CHAPTER 5. Conclusion and Future Scope 16-17

5.1 Conclusion 16

5.2 Future Scope 17

References vi

**LIST OF TABLE AND FIGURES**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Table Name** | **Page No.** |
| 2.1 | Pin List of 16x2 LCD | 10 |

|  |  |  |
| --- | --- | --- |
| **Figure no** | **Figure Name** | **Page no** |
| 1.1 | Drunk Driving | 1 |
| 1.2 | Result of drunk driving | 2 |
| 1.3 | Motor Vehicle Act | 3 |
| 1.4 | Motor Vehicle Bill | 4 |
| 2.1 | Arduino Nano | 6 |
| 2.2 | MQ-3 | 7 |
| 2.3 | DC Motor | 8 |
| 2.4 | 16x2 LCD | 9 |
| 2.5 | Arduino IDE | 11 |
| 3.1 | Circuit Diagram | 12 |
| 3.2 | Schematic Circuit | 13 |
| 3.3 | Alcohol Limit graph | 13 |
| 4.1 | Result with no alcohol detected | 14 |
| 4.2(a) | Start engine | 15 |
| 4.2(b) | Alcohol Below limit | 15 |
| 4.3(a) | Start engine | 15 |
| 4.3(b) | Alcohol above limit | 15 |
| 4.4(a) | Start engine | 15 |
| 4.4(b) | Alcohol above limit | 15 |

# ABSTRACT

# CHAPTER 1:

**INTRODUCTION**

With rapid transformation in the healthcare system and a growing need for quality healthcare, the hospital managers are now combining business expertise with an understanding of the healthcare system to increase efficiency and effectiveness of healthcare delivery and patient satisfaction.

The solution developed will mainly help the hospital managers & nurses. It will provide the ability to automate patient registration, schedule & search appointments. Nurses will do the initial diagnosis and they will check patient’s history, finally they will prescribe medicine and refer the patient to appropriate doctor. Nurses can also manage the medicine requirement of patients in this system & they can send prescription to stores.

Our Services uses an integrated approach, working with health care professionals across the industry to fill the current gap in health. Services uses technological advances to bring health care provision into the twenty-first century. Telemedicine is the use of medical information exchanged from one site to another via electronic communications to improve a patient's clinical health status.



Fig 1.1 **NAME OF THE IMAGE**

* 1. **AGILE METHODOLOGY**

Agile software development is an approach of software development in which requirements and solutions are a result of cumulative effort of self-organizing and cross-functional teams and their end-users. [1] The agile [2] methodology is a practice or a method in which the whole project is divided into small continuous iterations. Each iteration comprises of simultaneous development and testing of the project. It is unlike the waterfall methodology [3] where whole project is done all at once and the testing part is done at last. The Manifesto for Agile Software Development is built upon twelve basic principles: [4]

1. Providing customer satisfaction by timely delivery of product.
2. Always welcome changes, even in the end stages of project.
3. Frequent delivery of working software (weekly).
4. Daily standup meet between developer and business associates.
5. Projects are built around motivated and trusted individuals.
6. Face-to-face conversation.
7. The primary measure of progress is the working software.
8. Constant pace is to be maintained
9. Technical excellence and good design is paid continuous attention.
10. It is essential to maintain simplicity.
11. Self-organizing teams leads to best designs and architectures.
12. On a regular basis, team checks how to become more effective and makes the adjustments accordingly.

Although there are many differences between agile and waterfall model but one of them is the approach which they both use for quality and testing purposes. Waterfall methodology uses separate testing and build phase where build phase is followed by the testing phase. Agile methodology is famous for its simultaneous development and testing because here in each iteration testing is done. So if even a small error is found it is corrected simultaneously. Thus, it reduces the time and increases the efficiency at the same time.

* 1. **AGILE PROJECT MANAGEMENT ROLES:**

There is a team of many people which lend their hand for the successful completion of a project. Under agile methodology, the team comprises of following bunch of people in each team which play a prominent role in the team: [5]

1. **Product owner:** The person who is the link between customer, business stakeholders and the development team. The product owner is the main person who knows about the product and the customer’s needs and priorities. He daily checks and clarifies the developer team about the requirements from the customer’s end.
2. **Development team members:** Product is created by these people. It includes a software developer, designer, writer, data engineer, tester, etc. whosoever is required for that particular project and has expertise in the field of project.
3. **Scrum master:** He is the main support system of the developing team who ensures that the agile process remains consistent. He is responsible for clearing the roadblocks from the organization’s side.
4. **Stakeholders:** These people provide valuable feedbacks that are importantbut are notultimately responsible for the product and also get affected by the project’s outcome.
5. **Agile mentor:** An experienced person who is good at implementing agile methodology into project and share his experience with the project team. He is a guide who gives his valuable knowledge and shares his experience about the project he has worked in past and advices the team for different approaches and techniques.
   1. **AGILE PROJECT MANAGEMENT EVENTS**

Although there are many stages in the project development but when we follow agile methodology mainly seven events are included in project development. These can be explained as follows: [5]

1. **Project planning:** It is the initial planning of the project where roadmap of the project, product vision is done.[5]
2. **Release of planning:** Planning for the next bunch of product features to be released and only one release date is planned at a time.[5]
3. **Sprint:** Also known as the iterations, is a time duration in which team creates a functional product. This time duration generally lasts from one to four weeks, but cannot be larger than four weeks. All the sprints must be of the same duration.[5]
4. **Sprint planning:** Before the start of eachscrum **a** meeting is done so that all the work to be done in that scrum can be planned and the team can commit to its goals.[5]
5. **Daily scrum:** Each day in the morning a 15 minutes meetup is held by the development team members to know each other’s work to be done in that particular day, the work that is lagging and the work to be done in the future. Daily scrum enables the team to know whether they are having any hindrances in their path or not and how to improve their efficiency in any particular area.[5]
6. **Sprint review:** After each sprint is completed a meeting is held by the product owner where the development team shows how much part of the planned work is done and how much is left so that it must be included in the next sprint i.e. by updating the product backlog.[5]
7. **Sprint retrospective:** At the end of each sprint the scrum team will evaluate their methods and procedures and look forward to the changes they can implement for clearing the roadblocks so that the functionality of the project can be implemented.[5]

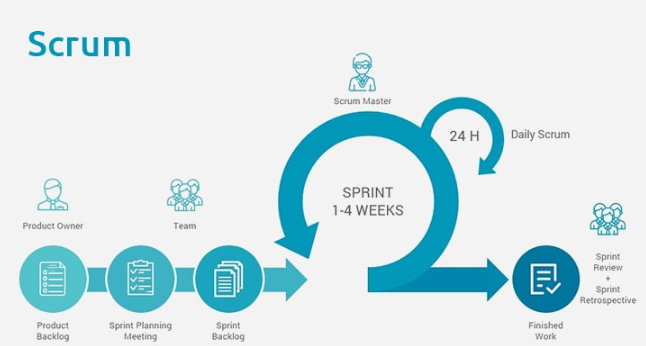


Fig 1.2 Scrum Process

**CHAPTER 2:**

**BUSINESS REQUIREMENT SPECIFICATION**

|  |  |
| --- | --- |
| Front End | HTML5, CSS3, JavaScript |
| Middleware | Java (Servlet, JSP,WEB API2,MVC ) |
| Backend | Oracle |

Table 2.1: Technologies used

|  |  |  |
| --- | --- | --- |
| Technology | Hardware | Software |
| Java | Desktop PC with 8GB RAM | 1. Eclipse IDE for Java EE Developers (Oxygen)  2. Tomcat 9  3. MySQL Community Server 8.0  4. Oracle 11g express version |

Table 2.2: Hardware and Software required

* 1. **High Level Business Requirement**

Primary focus is to complete developing the critical requirements and then to proceed with the remaining requirements.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No.** | **Business Requirement ID** | **Short Description** | **Description in detail** | |
| 1 | Req\_1 | Hospital and users Registration | | Ability of the system to procure the fundamental details of the Hospital and users. |
| 2 | Req\_2 | Hospital and users Authentication | | Ability of the system to authenticate the credentials of the registered Hospital and users |
| 3 | Req\_3 | View Hospitals | | Ability of the system to  display hospitals |
| 4 | Req\_4 | View Doctors | | Ability of the system to display the details of the doctors of a selected hospital |
| 5 | Req\_5 | Book appointment | | Ability of the system, which allows the user to book an appointment of the selected doctor. |
| 6 | Req\_6 | View Patient details | | Ability of the System, which allows a particular hospital to view the details of the patients who booked appointment of its doctors. |

Table 2.3: High Level requirements

* 1. **Process Architecture**

Below is the overall functional flow of the project including the components of interaction.

****

Fig 2.1: Functional Flow of Project

* 1. **Detailed Business Requirement**

The functional requirements are charted for each of the high-level requirements.

The following elements are captured for each business requirement in the table provided below:

1. **Requirement Type**

F Core Functionality

E Exception

UI User Interface

R Reporting

1. **Priority of Requirement**

1=Base Functionality

2=Advanced Functionality

3=Additional Opportunities

1. **Originator**

Name of the business process of the system/ department or function name in the customer organization

The Requirements in this document are prioritized as follows:

|  |  |  |
| --- | --- | --- |
| **Value** | **Rating** | **Description** |
| 1 | Critical | The requirement is critical to the success of the project. The project will not be considered complete without this requirement. |
| 2 | High | This requirement is high in priority, but the project can be implemented at a bare minimum without this requirement. |
| 3 | Medium | This requirement is somewhat important, as it provides some value but the project can proceed without it. |
| 4 | Low | This is a low priority requirement or a “nice to have” feature if time and cost allow it. |
| 5 | Future | If any future work is to be included |

Table 2.4: Prioritized List of the Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Req. #** | **Rationale Categorization** | **Business Requirement** | **Req. Type** | **Priority** |
| Req\_1.1 | User  Registration | When the User clicks on the registration link, it should re-direct to volunteer registration form. | UI | Critical |
| Req\_1.2 | Hospital  Registration | When the Hospital clicks on the registration link, it should re-direct to organization registration form. | UI | Critical |
| Req\_1.3 | User  Registration | User needs to fill some of the basic attributes/fields as mentioned below in requirement: First Name, Last Name, Age, Gender, Contact Number, E-mail, Password, Volunteer Id (auto generated and hidden from UI) | UI | Critical |
| Req\_1.4 | User  Registration | Clicking ‘Submit’ should validate the datatype constraints for each field | F | Critical |
| Req\_1.5 | Volunteer  Registration | Volunteer failing to provide information on the mandatory fields be provided with an alert message – ‘Please update the highlighted mandatory field(s).’ Also, highlight the missed out field in red | E | Medium |
| Req\_1.6 | User  Registration | Post-successful field level validation, save the information in the database | F | Critical |
| Req\_1.6 | User  Registration | Upon saving the information in the database, display the message ‘Your details are submitted successfully’. | E | Medium |
| Req\_1.7 | Hospital  Registration | Hospital needs to fill some of the basic attributes/fields as mentioned below in requirement: Name of the organization, Contact Number, E-mail, Password, organization Id (auto generated and hidden from UI) | UI | Critical |
| Req\_1.8 | Hospital  Registration | Clicking ‘Submit’ should validate the datatype constraints for each field | F | Critical |
| Req\_1.9 | Hospital  Registration | Provide organizer failing to provide information on the mandatory fields with an alert message – ‘Please update the highlighted mandatory field(s).’ Also, highlight the missed out field in red | E | Medium |
| Req\_1.10 | Hospital  Registration | Post-successful field level validation, save the information in the database | F | Critical |
| Req\_1.11 | Hospital  Registration | Upon saving the information in the database, display the message, ‘Your details are submitted successfully’. | E | Medium |
| Req\_2.1 | User Credential Authentication | A registered volunteer – is able click ‘Login’ link, after keying in ‘contact number’ & ‘Password’ field and get his credentials authenticated with the existing database entry. | F | Critical |
| Req\_2.2 | Hospital  Credential Authentication | A registered Hospital – is able click ‘Login’ link, after keying in ‘contact number’ & ‘Password’ field and get his credentials authenticated with the existing database entry. | F | Critical |
| Req\_3.1 | Patient Details | When a hospital is signed in they can see the patients who booked an appointment of its doctors. | UI | Critical |
| Req\_3.1 | Hospital Details | When a user is signed in he should be shown the list of hospitals | UI | Critical |
| Req\_4.1 | Doctor details | When a user is selecting a hospital he can see list of doctors of that hospital with a book appointment option. | UI | Critical |
| Req\_5.1 | Book Appointment | When user clicks appointment  his details along with the doctor details should be stored in data base. | F | Critical |
| Req\_6.1 | Patient Details | When a hospital is signed in they can see the patients who booked an appointment of its doctors. | UI | Critical |

Table 2.5: Detailed Business Requirement

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Field Type** | **Data Type** | **Mandatory** | **Possible Values** |
| First Name | Text(50) | Alphabetic | Yes |  |
| Last Name | Text(50) | Alphabetic | Yes |  |
| Age | Numeric(2) | Numeric | Yes |  |
| Gender | Drop Down | NA | Yes | Male, Female |
| Contact Number | Text(10) | Numeric | Yes |  |
| E-mail | Text(50) | Alphanumeric | No |  |
| Password | Text(15) | Alphanumeric | Yes |  |
| Volunteer Id | Auto-generated(5) | Numeric | Yes | Non-editable system generated text |
| Name of the Hospital | Text(50) | Alphabetic | Yes |  |
| Contact Number | Text(10) | Numeric | Yes |  |
| E-mail | Text(50) | Alphanumeric | No |  |
| Password | Text(15) | Alphanumeric | Yes |  |
| Hospital Id | Auto-generated(5) | Numeric | Yes | Non-editable system generated text |
| Name of the Doctor | Text(50) | Alphabetic | Yes |  |
| Doctor specialization | Text Area(rows-5,cols-10) | Alphanumeric | Yes |  |

Table 2.6: Data Types of the field required

**CHAPTER 3:**

**USER INTERFACE SPECIFICATION**

Use case document is to systematically capture requirements for the project and the system to be developed in terms of use cases. Functional use cases are captured in this document.

It also serves as the input for the project scoping.

The scope of this document is limited to addressing the use cases from a user, quality, and non-functional perspective.

* 1. **Home Page**

User interface deals with the capture of user details. The user here shall be the operator of the system and will be keying in the patient or doctor information into the system.

User being able to access the homepage & get redirected to the ‘Patient Registration’ or ‘Hospital Registration’ page upon click of ‘Register’ link.

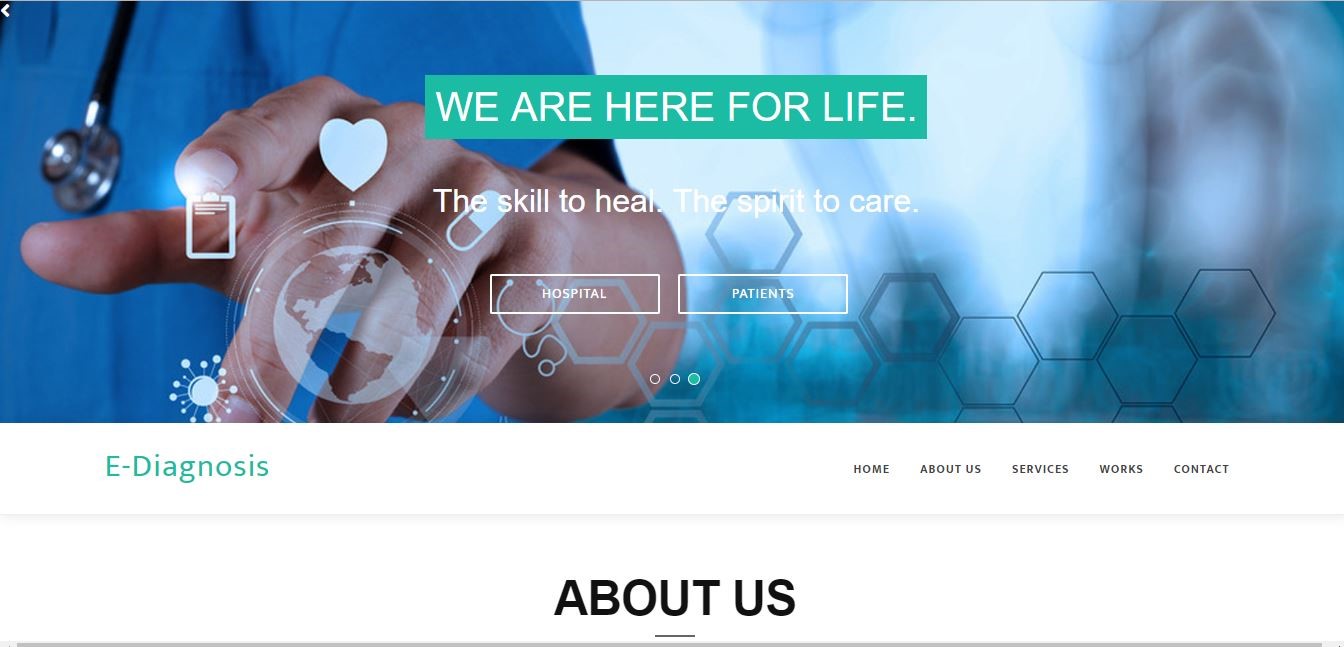


Fig 3.1 (a) Homepage

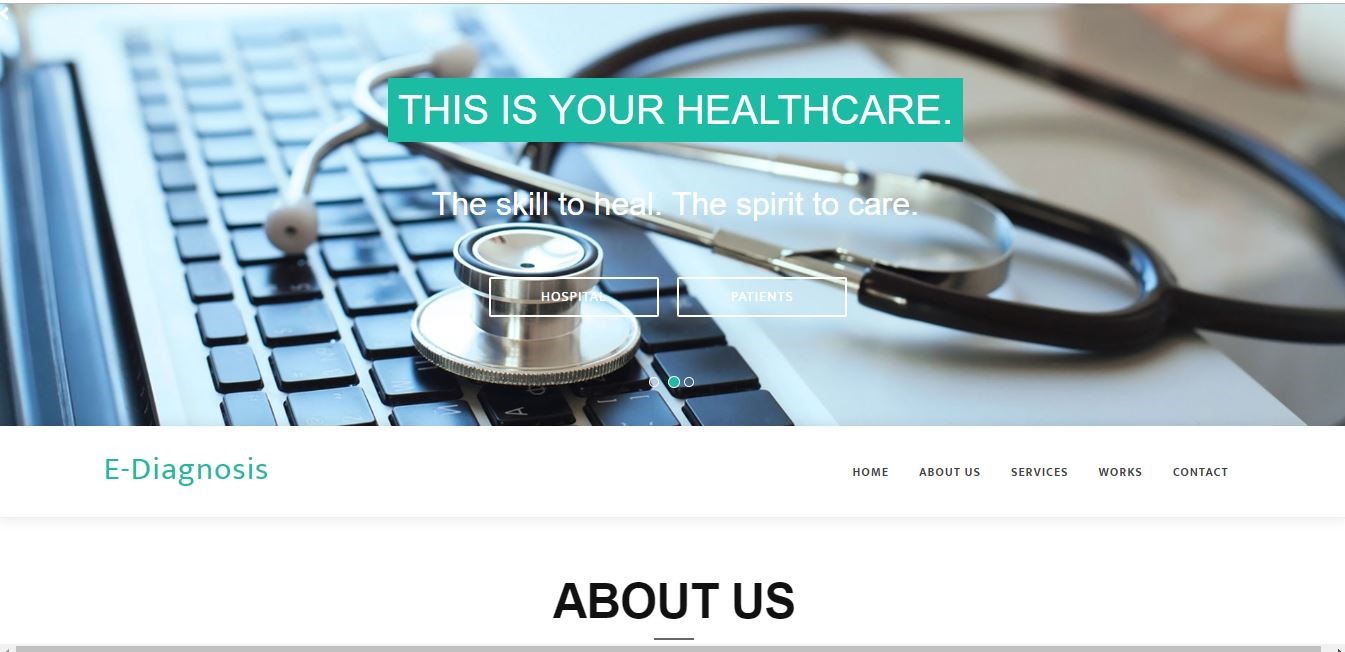


Fig 3.1(b): Homepage

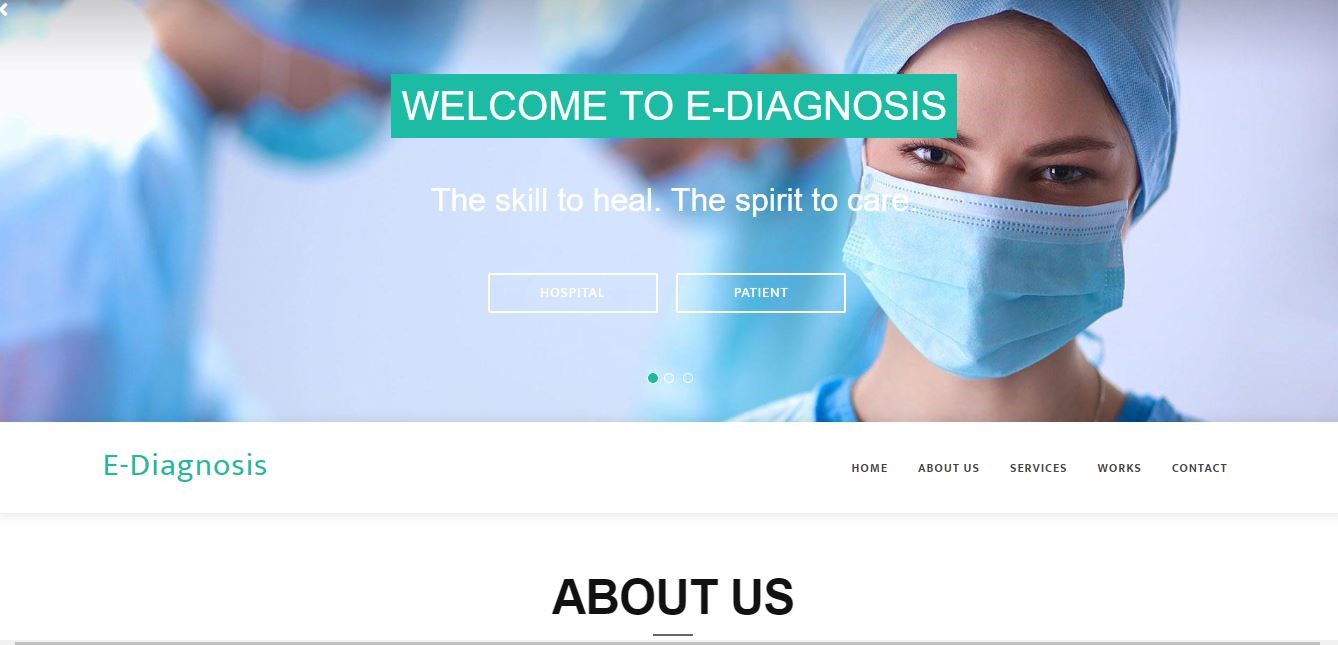


Fig 3.1(c): HomepageFig 3.2 About us – Homepage

* + 1. **Services - Homepage**

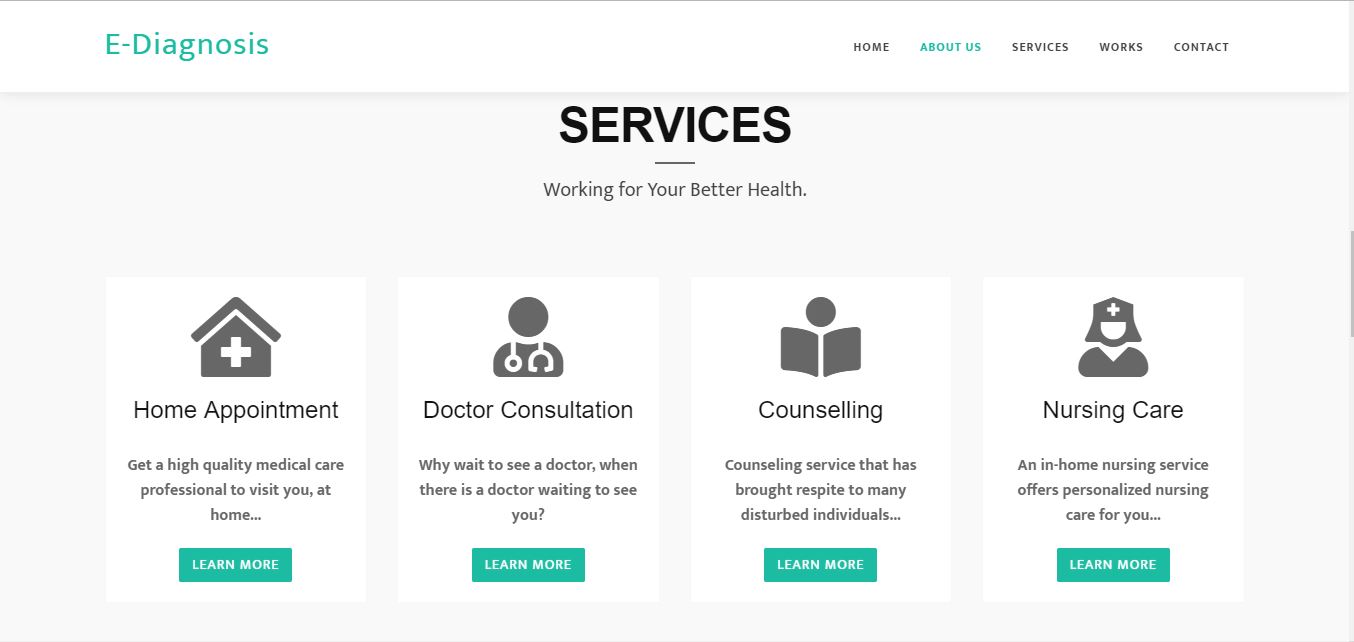


Fig 3.3(a) Services – Homepage

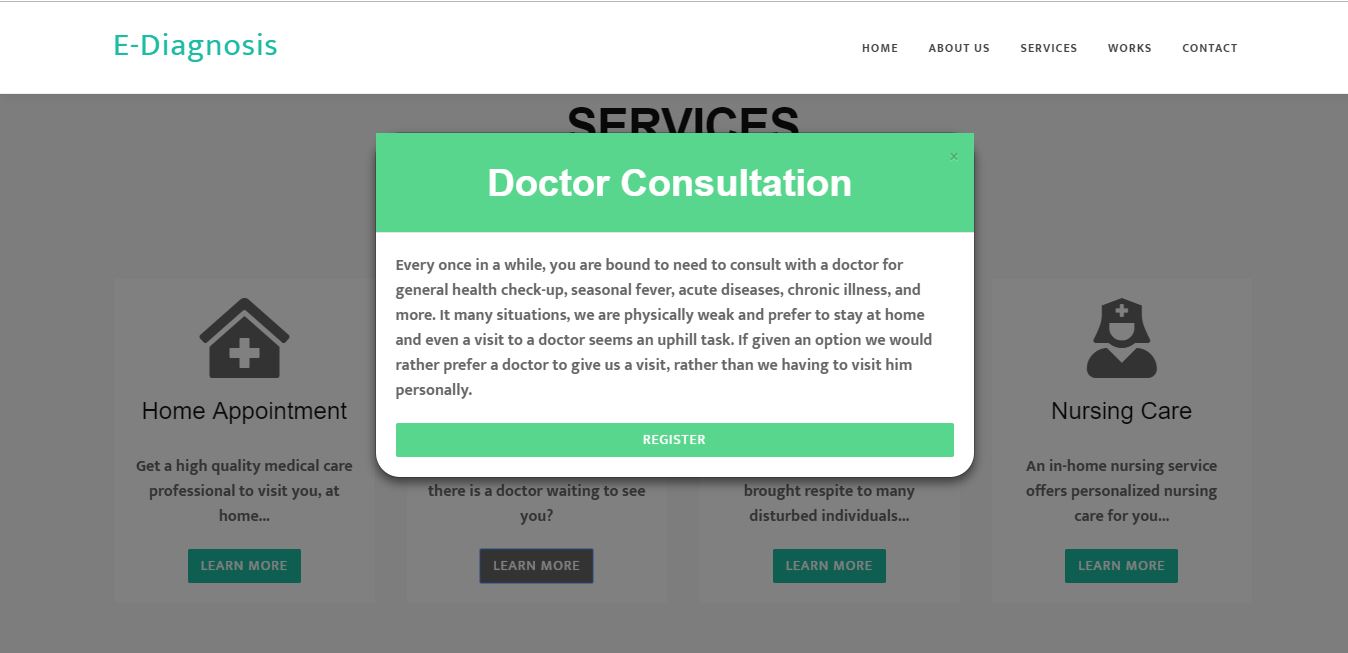


Fig 3.3 (b) Services - popup

* + 1. **Portfolio**

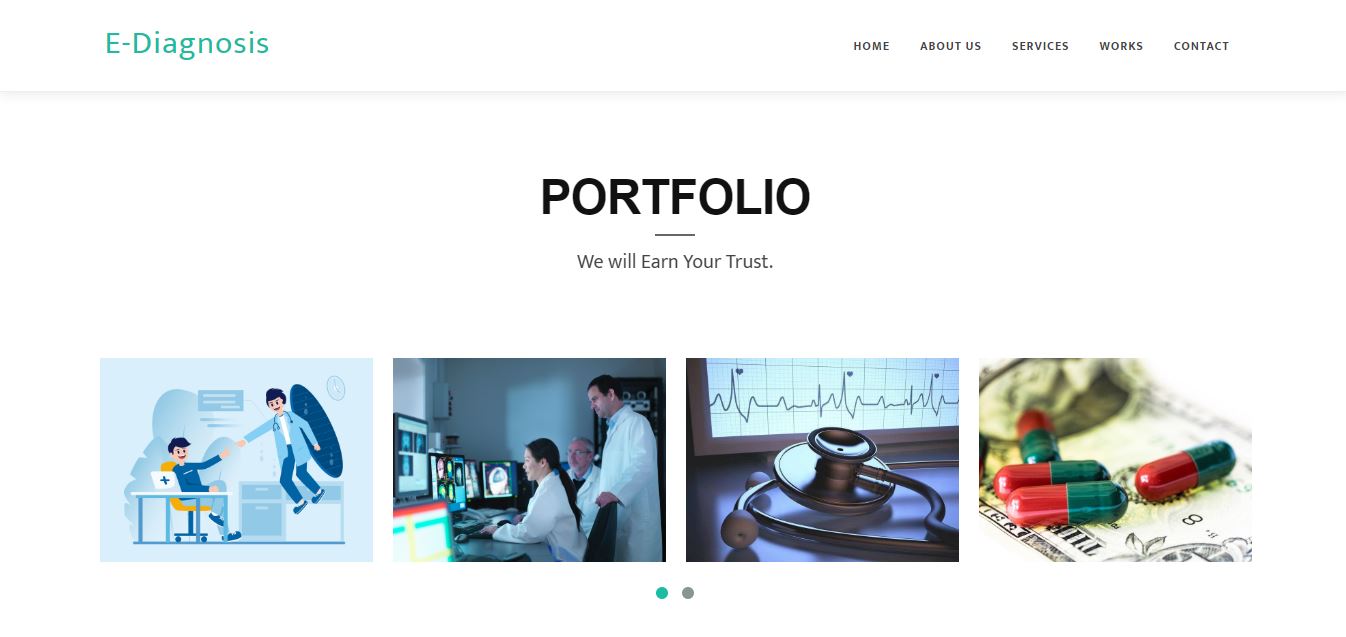
****

Fig 3.4 (a) Portfolio

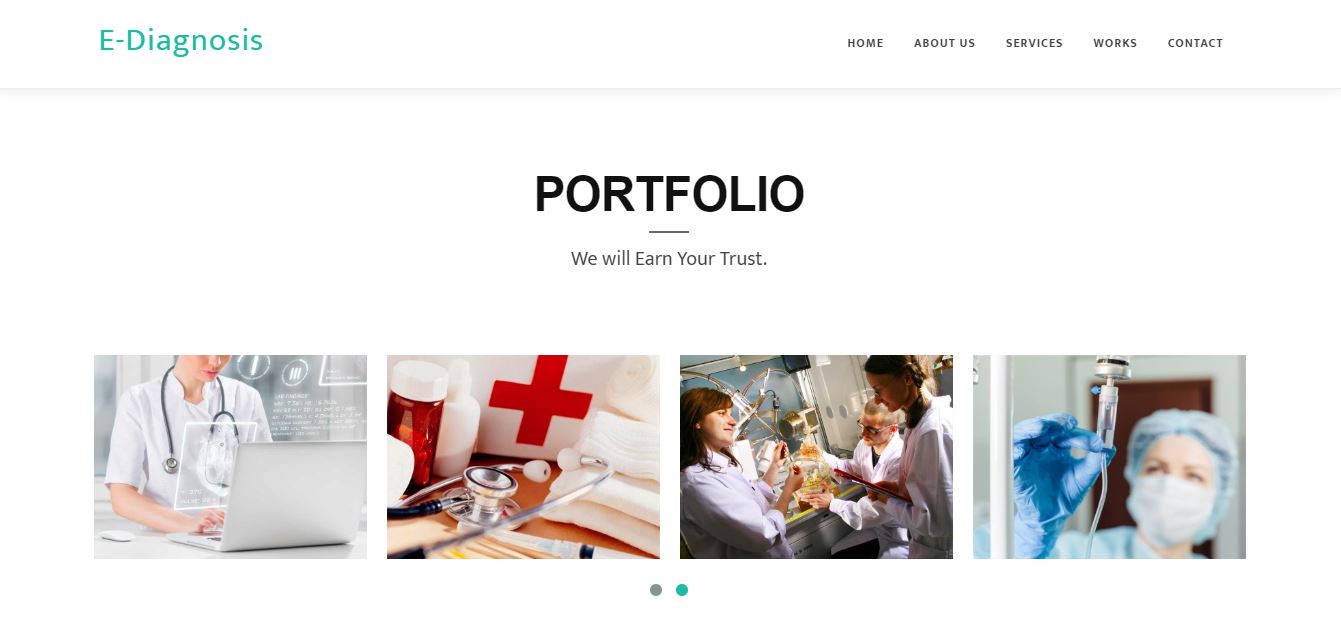


Fig 3.4 (b) Portfolio

* + 1. **Contact Us**

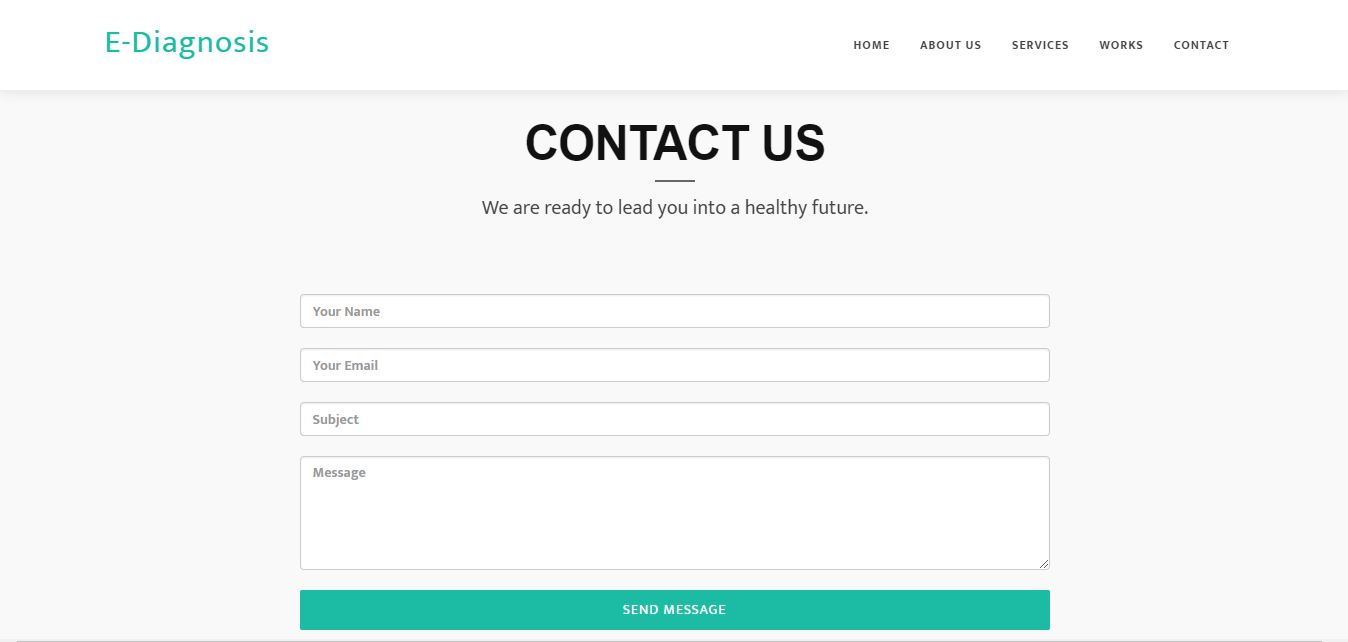
****

Fig 3.5 Contact us – homepage

* 1. **Login Page**

User being able to access the homepage & get redirected to the ‘Login page’ upon click of ‘Hospital’ or ‘Patient’ link.

A new user should be able to Register itself and be able to provide his details and get enrolled in the system.

* + 1. **Business Rules**

Business rules are defined using the following attributes:

* When the user clicks on the registration link, it should re-direct to registration form.
  + - User needs to fill some of the basic attributes/fields, First Name, Last Name, Age, Gender, Contact Number, User Id, Password
* Clicking ‘Submit’ should validate the datatype constraints for each field
* Post-successful field level validation, save the information in the database
* Upon saving the information in the database, display the message, “Your details are submitted successfully”.

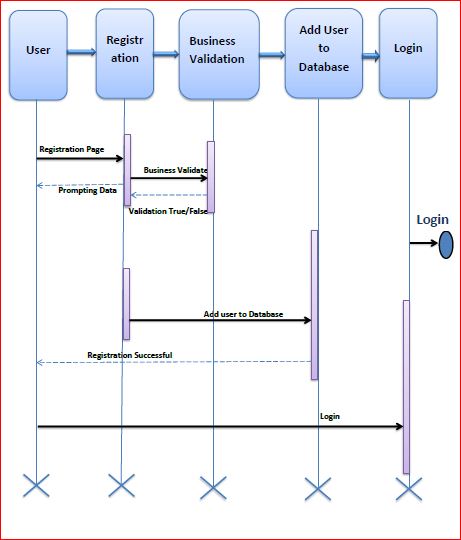


Fig 3.6 Use Case Model

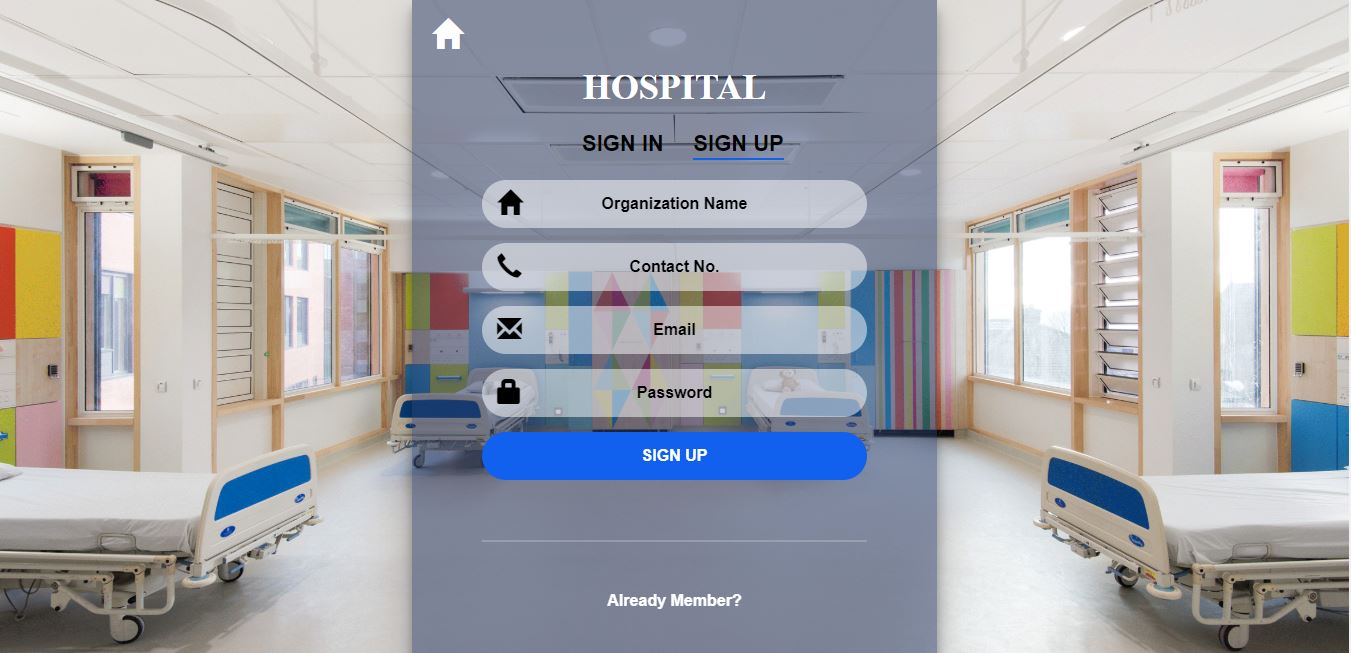
****

Fig 3.7 Hospital Sign Up

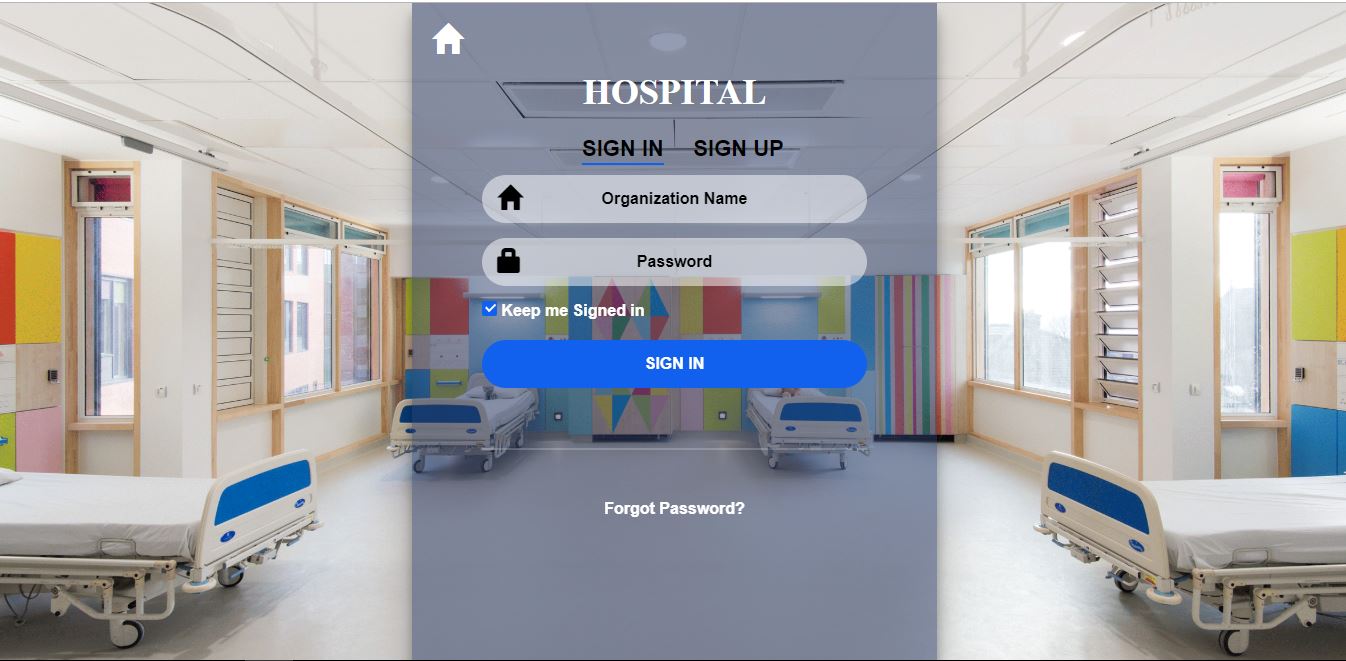


Fig 3.8 Hospital Sign In

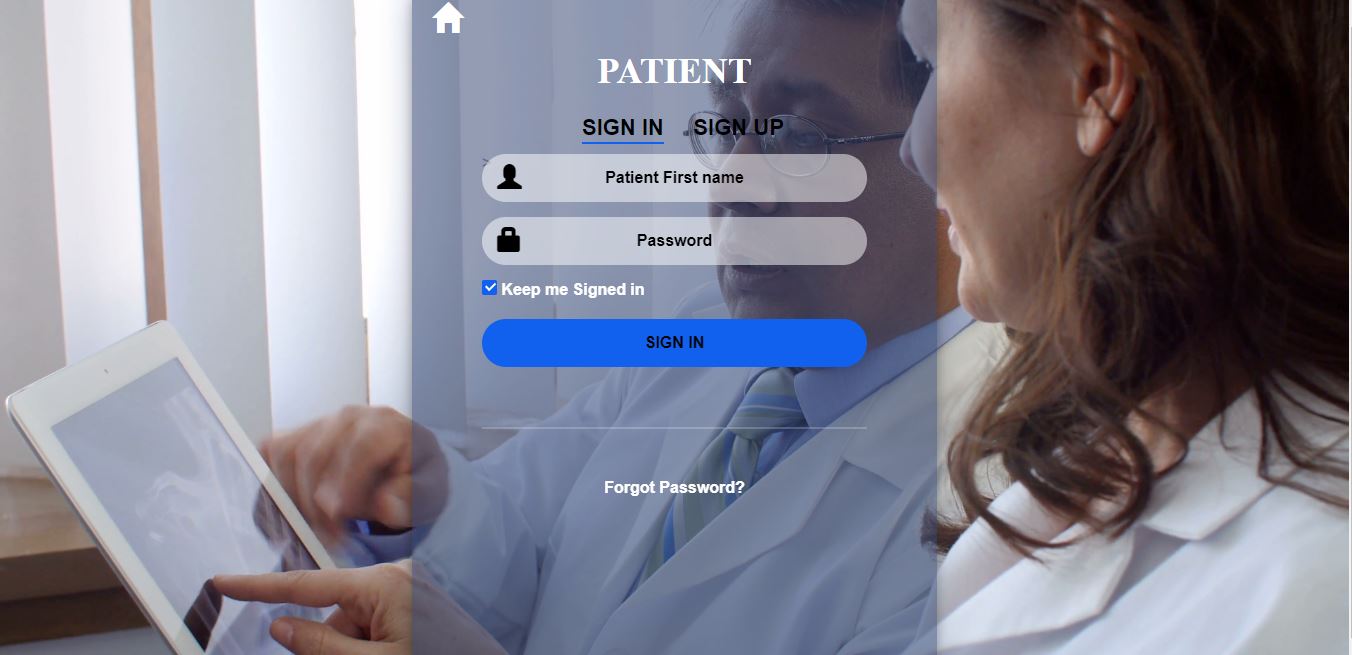


Fig 3.9 Patient Sign In

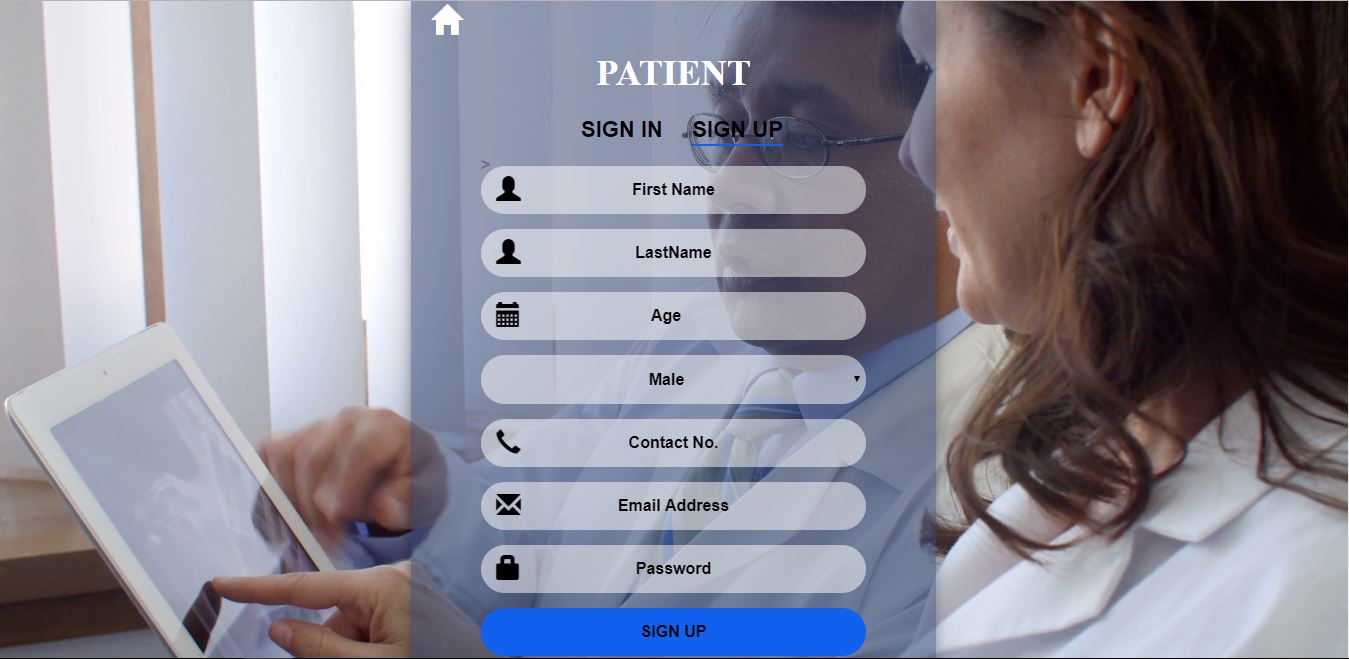


Fig 3.10 Patient Sign Up