**CHAPTER ONE**

**INTRODUCTION**

**1.1 Background of the Study**

Different Nigerian Polytechnic Institutions admit thousands of students every session and a major criteria of these students’ registration is to undergo medical tests. The importance of this initial medical test cannot be over emphasized, from providing health status to creating a baseline health database which can be consulted during emergencies, this test also ensures these prospective students do not pose any health challenge to other pupils of the institution but as the intake every year increases there is need for a convenient method of conducting these tests for the newly admitted students. The implementation of the web based medical test scheduler and result issuing system is to speed up the process of registration at these institutions of higher learning while solving the need to automate the most processes and move away from manually having to visit the medical centers only to book appointments for their said medical tests. Over the years this has shown to be time consuming students and wastage of money as they travel sometimes to set these appointments only to fail to meet the stipulated time.

**1.2 Statement of the Problem**

In past years this medical tests have been shown to be very tedious on both the students and the medical team performing the tests and because of the huge student’ information in the health care, it is not practical to keep using the paper documents for tracking and managing the student medical test schedules and because these tests are paramount for student registration, it creates a situation with long queues and no sense of order or control with some students missing lectures or classes just to have their medical test done. All these difficulties in the system prompted for a more convenient procedure that would not only help the students but provide a sense of order for the medical staff as well. The traditional way of handling the medical tests of these students have shown not to be so convenient anymore, this project undertakes the challenge to produce a more innovative measure to tackle this problem.

**1.3 Aim and Objectives of the Study**

The aim of the research work is to develop a Web-Based Student Medical Test Scheduler and Result Issuing System to eliminate student waiting time during registration.

**Objectives:**

The objectives are stated below:

1. To design a web based medical test scheduler that enables admitted students schedule online for their medical tests conveniently.
2. To implement this system that also serves as a result issuer.
3. To evaluate the benefit and the advantages this system will have over the pre-existing traditional way of scheduling these medical tests.

**1.4 Scope of the Project.**

This research work is centered on the development of a favourable algorithm for scheduling medical tests for admitted students while allowing the students use the system online, book and confirm their own appointment by themselves, this system also provides a dependable medical database as well as provide results for the medical tests conducted, printable online, it also focused on the development of a user-friendly, web-based interface to serve its purpose.

**1.5 Limitations of the Study.**

This study's scope has been constrained by several core issues, including:

**Inadequate information –** This factor is the leading limitation of this research work as it put a bar on how far the research work could go.

**Time** - Time also proved to be a limiting factor due to life, family, school, work and ‘side hustle’ as a baker competed for this researcher’s time.

**Access to literature** – while some materials came easy to access others were restricted.

**Finance** – Some material that would have greatly facilitated the research was restricted to payment before access which prevented the researcher from gaining information that would have been of immense value to this research work.

**1.6** **Significance of the Study.**

This web-based student medical scheduler and result issuer system ensures that a convenient and an easy means of scheduling appointments. Students can then be batched into groups that are served conveniently based on appointment. This scheduler system provides an electronic document instead of using the traditional method (paper document) which made it uneasy to document and retrieve student information. On this note, the importance of a medical test scheduler and result issuing system cannot be over emphasized as it not provides the flexibility, and supports remote access for these students but also provides a well-managed medical database for these students.

Scheduling is an essential task in any medical line. This system organizes student appointments in flexible schedules.

**1.7** **Project Organization**

The research work is made up of five chapters and are outlined thus:

**Chapter One: Introduction**

This chapter introduces, the background of the study, the statement of the problem, the aim and objectives, the scope of the study, limitations of the study, the significance of the study, project organization, and the definition of terms.

**Chapter Two: Literature review**

This chapter focuses on the literature review, and the contributions of other scholars on the subject matter being discussed.

**Chapter Three: Methodology and Design**

Chapter three is concerned with the presentation of the results of system analysis and design. It presents the research methodology used in the development of the system to facilitate an understanding and effective future implementation of the system.

**Chapter Four: System Implementation Evaluation**

This chapter describes the system implementation and documentation, analysis of modules, and system requirements for implementation.

**Chapter Five: Summary, Conclusion, and** **Recommendation**

The chapter provides a summary of major findings, conclusions, and recommendations based on the study conducted.

**1.8 Definition of Terms**

1. **Project:** A project is a collection of activities that must be accomplished over a set period while keeping cost and other constraints in mind to achieve a specified conclusion or objective.
2. **Algorithm**: An algorithm is a set of instructions or steps used to solve a problem or accomplish a task. Algorithms are used in computer programming to describe a set of operations that need to be performed for a computer to complete a certain task.
3. **System**: A system is a set of elements or components that are organized for a common purpose. It can refer to a set of components that interact to form a complex whole, or to a set of procedures or rules that are followed to achieve a specific goal.
4. **Database**: A database is a system intended to easily organize, store, and retrieve large amounts of data. It consists of an organized collection of data for one or more uses, typically in digital form.
5. **Web**: A subset of the internet consisting of pages that can be accessed by a web browser.
6. **Web based system**: Software hosted on a remote server on the web, which can be accessed by users using a web browser.

**CHAPTER TWO**

**LITERATURE REVIEW**

**2.1 Introduction**

This chapter aims to demonstrate how the problem under research is connected to previous research, current practice, or other domains of knowledge by citing relevant publications by other researchers that have addressed a similar issue. Furthermore, this chapter will include a synthesis of existing research on the issue, emphasizing areas of agreement, disagreement, and gaps in the literature, in order to illustrate the relevance of the project topic in the field and to propose chances for future study.

**2.2 Literature Review**

Sanjeevani et al. (2018). Doctor Appointment Online Booking System. If a person becomes unwell and wishes to see a doctor for a check-up, he or she must go to the hospital and wait until the doctor becomes available. While waiting for an appointment, the patient simultaneously waits in line. If the doctor cancels the appointment due to an emergency, the patient will not be aware of the cancellation until he or she enquires about the delay. Because mobile communication technology is continuously evolving, one may employ mobile applications to address such challenges and discomfort for patients.

Furthermore, the presented work in this article is an Online Hospital Management Application that employs an Android platform to make scheduling a doctor's appointment simple and dependable for users. The architecture is designed to allow users to access the booking system via a portable computer system, a desktop computer system, or a mobile phone as a web browser. We employed client-server architecture and thin client-server. The application's front end is Asp.net, while the back end is a SQL database.

In conclusion, patients may now easily book and manage their own appointments thanks to the introduction of a web-based NHIS medical appointment booking system. They will be reminded of their appointments via SMS/email, which will be sent to them as soon as possible before their appointment date. On the Home page, the system also displays a fast glimpse of their appointment. These capabilities may indirectly assist to minimize the amount of missed medical appointments and no-show patients. The technology may also assist to lessen the workload of healthcare staff by allowing them to simply produce medical reports. Because of technical limitations, the system cannot show bio-data such as X-rays and laboratory findings.

Ele et al. (2020). A Web-Based Medical Appointment Scheduling with SMS Alert Notification System. Long wait periods for registering and arranging appointments with doctors are common in Nigeria, particularly at government institutions. The inability to make appointments remotely, as well as the lack of a clearly-defined alert notification system to remind patients and physicians of their appointments, inadequate and permeable security mechanisms for protecting patient personal information, long waits in lines and waiting rooms to see a doctor, as well as the inefficiency and high expense of health care delivery and services. In addition, missed healthcare visits are the leading source of avoidable incompetence that affects a patient's well-being and medication outcomes because there is no apparatus in place to notify patients when appointments are postponed or canceled. To address these concerns, a web-based medical appointment scheduling system with SMS alert notification is proposed, with the University of Calabar Teaching Hospital (UCTH) serving as a case study.

Moreso, to collect information on the current system, many strategies were utilized. In this study, information was gathered through interviews, questionnaires, the internet, literature, and requirements determination. The Structured System Analysis and Design Methodology (SSADM) was used for analysis and design in this study. This style is used because it gives users or software developers with bright, easily understandable documentation that includes multiple graphic representations of the system. The database for this study was designed using MySQL, and the proposed system was constructed using the PHP programming language.

In conclusion, the developed system can eliminate missed healthcare visits and gives patients a notification when appointments are stopped or canceled. It provides a platform for remote appointment booking and enough protection to secure patient medical information since each registered patient and physician has a unique username and password for login.

Noorsyahira et al. (2017). Medical Appointment Application. The main issue addressed in this paper is how to overcome the limitations of the current manual medical appointment system, such as an increase in the number of calls for an appointment and a morning rush for an urgent appointment, as well as an increase in the need for extra reception staff due to the long queue in the Parit Raja and Batu Pahat areas.

Furthermore, the prototype model was employed in the application process. The hardware and software utilized to construct this system are MySQL Database and PHP and JavaScript programming languages. Although phpMyadmin software is used to manage the database, the application's interface is designed using Bootstrap.

In conclusion, the Medical Appointment application is a web-based mobile application designed to aid in patient scheduling with Batu Pahat and Parit Raja offices. Some advantages have been recognized based on the created application. The application has compiled a list of medical practices in the Batu Pahat and Parit Raja areas. The user, the patient, may quickly arrange an appointment with the practice of their choice, avoiding a huge line at the clinic. The application is designed to bring together any medical practices that choose to join the application, with a focus on practices in the Batu Pahat and Parit Raja areas. It is also to assist the patient in avoiding any complications in scheduling appointments with these clinics.

Hema (2019). Effective Online Medical Appointment System. Direct paper appointments make it tough to handle patients. Healthcare has been the most significant service in many industrialized countries during the previous two decades. It is difficult to obtain appointments by contacting the hospital directly and waiting in line. The fundamental idea behind this project is to make it easier for people to get appointments by using an internet application. The effort to the patients will be decreased with this application, which contains the details of the doctor and their available time, and time will be saved for both physicians and patients. The doctor is free to plan his own time using the proposed system.

Moreso, in the proposed system, doctors and patients are brought together in one location that is managed by the admin, and the admin allows patients to register and search for doctors based on their location, and patients can book by selecting time slots, and the admin will confirm the booking, so everything is computerized and done quickly, saving time.

In conclusion, this web-based application solves the problem of organizing and booking appointments depending on the user's preferences. Here the customer may pick doctors by checking their details and reviews. As a result, this project provides an effective solution in which customers may examine available booking slots and pick the ideal day and time. The already reserved slot will be marked yellow and will be unavailable to anybody else for the duration of the reservation. Users can also cancel their reservations at any moment using this method. The doctor can use this application to notify himself of his own schedule. Hospitals can simply manage their registration and appointment processes, as well as track patient flow to the doctor. And, with this application, both the doctor and the patient may save time.

Cara (2017). Online Medical Appointment Scheduling System. The high degree of competition in the provision of health services necessitates not only the provision of high-quality health care but also an ongoing collaboration with patients, who are the primary customers of health facilities. There are several issues with manual appointment scheduling, including as:

1. **Constant phone calls that reduce productivity**: Phone calls might keep a receptionist busy and prevent her from attending to patients who are waiting in the lobby. This means that in order to provide a competent service, at least two receptionists are required, which might be rather costly for small clinics.
2. **Office hours are limited**: Potential clients who can only attend or phone a clinic at a specific time so there is an increase in the loss of consumers if the clinic is not open at that time.
3. **More personnel required**: More personnel (particularly receptionists) may be required, resulting in a rise in monthly expenditures. Small and medium-sized medical clinics may struggle with such a rise in costs. As a result, the proposed system

However, during the construction of this online clinic website, it is extremely likely that a customer would want to add some additional requirements therefore agile development approach is favored and was implemented; all of the software and systems that we used are based on what WordPress uses. The most common web server “phpMyAdmin” was integrated, which allows customers to administer their databases using a web-based GUI.

In conclusion, the researchers believed that all of the requirements/objectives were met. We can state that the development team learned various strategies throughout the construction of this website, allowing it to be improved and maintained with the required efficiency.

**2.3 Summary of Related Literature Reviews**

|  |  |  |
| --- | --- | --- |
| **Author & Year** | **Title & Description** | **Merit and Demerits** |
| Sanjeevani et al. (2018) | Doctor Appointment Online Booking System.  This is a web-based application that solves the problem of maintaining and arranging appointments depending on the user's preferences or expectations. | The system allows users to cancel their reservations at any time.  The system cannot show bio-data such as X-rays and laboratory results |
| Ele et al. (2020). | A Web-Based Medical Appointment Scheduling with SMS Alert Notification System.  The developed system will enhance appointment scheduling in hospitals by simplifying patients' and physicians' work and eliminating missed appointments with the aid of the remainder component. | The system saves the user time by eliminating the need to negotiate with the receptionist for a convenient appointment time.  The developed application is not of a top-notch quality |
| Noorsyahira et al. (2017). | Medical Appointment Application.  Medical Appointment Application is a web-based mobile application developed for handling appointment-booking processes for a few medical institutions in the Parit Raja and Batu Pahat area, regardless of the sort of service they schedule. | The system provides an opportunity for customers to chat with sellers directly.  The system lacks shipment management. |
| Hema (2019). | Effective Online Medical Appointment System.  This is a web-based application that solves the problem of organizing and booking appointments depending on the user's preferences. | The system is flexible as users can cancel their appointment at any time.  Difficult to find a date that suits both the patient and the doctor. |
| Cara (2017). | Online Medical Appointment Scheduling System.  The developed system enables existing and prospective patients to quickly schedule appointments with the clinic's many doctors. 365 days a year, 24 hours a day. | The technology enables online appointment administration and scheduling.  The system lacks an online payment feature. |

**2.4 Analysis of the Existing System**

The existing system which is the current system is manually operated, It is a yardstick that each student that gains admission into the school has to do a medical fitness test, due to the influx of students, the activity has been tedious for the medical laboratory workers, and the student, the student has to leave their homes as early as possible to queue for them to be among the first 100 student that will be attended to for that day.

The existing method of scheduling has a number of drawbacks. First, it can be time-consuming for both the student and the laboratory workers. students may have to wait on hold for a long time before being attended to.

Another drawback of the existing method is that it is prone to errors considering the influx of students, the test result may be assigned to the wrong person.

In conclusion, the existing method of medical test scheduling is a time-consuming and error-prone process that can lead to delays and inconvenience for students. An electronic scheduling system would be a more efficient and reliable method of scheduling tests.

**2.5 Analysis of the Proposed System**

The proposed system aims to investigate the advantages and benefits of using an online platform for scheduling medical tests for students. The research will focus on the convenience, cost-effectiveness, and ease of use of the application.

In past years these medical tests have been shown to be very tedious for both the students and the medical team performing the tests. Being that every student given admission needs to be tested, this situation creates long queues with no sense of schedule whatsoever. This web-based student medical scheduler and result issuance system ensures a convenient and easy means of scheduling appointments for these students based on a first come, first serve. Students can then be batched into groups that are served conveniently based on appointment. You can easily schedule and fix appointments and medical tests for the students with just a few mouse clicks.

One of the main advantages of the proposed system is convenience. Instead of an overcrowded environment and long wait lines, the student can know his or her test date and come on the scheduled date.

The system can also provide online updates to students on their test results, which can be accessed through the student portal of the system and downloaded by the student.

In addition to benefitting students, the web-based medical test scheduling system can also help laboratory personnel and clinics in streamlining their operations.

**CHAPTER THREE**

**METHODOLOGY AND DESIGN**

**3.1 Introduction**

A methodology is a formal study or inquiry, especially to unveil new facts or information; thus, research methodology should be good enough to enable the attainment of the specified objectives, which are accomplished through the use of specific components such as data collection and design requirements, and system modeling (use case, activity, and class diagrams). This chapter describes the input/output specifications as well as the system requirements for the currently under-construction web-based student medical test scheduling and result issuance system.

**3.2 Methods of Data Collection**

Before constructing any system, collecting data and facts about the existing system is necessary to comprehend what is going on. Two approaches were used in this study.

i. Observation of the Work Environment

ii. Documentation

**3.2.1 Observation of the Work Environment**

This approach was used to collect information and data for this study by observing how the manual system functioned. Detailed inspection revealed the most obvious weaknesses in the present system. The setting in which the observation is made can be altered in a variety of ways when using the observational approach.

**3.2.2 Documentation**

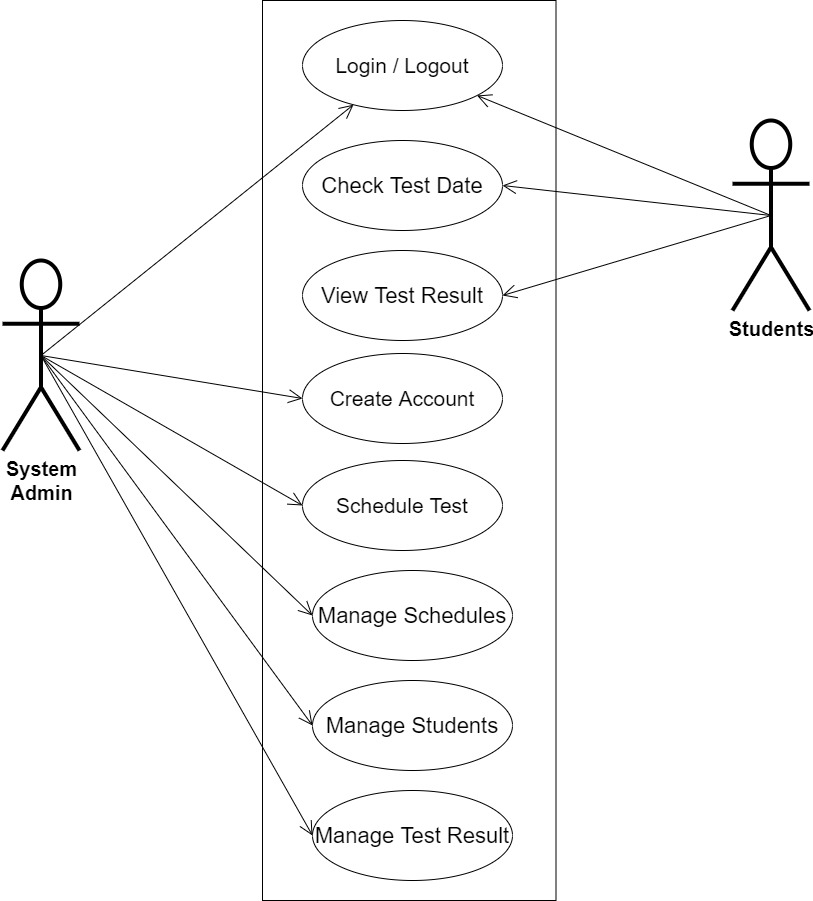
Documentation is a sort of secondary data collection. This strategy takes the use of journals, manuals, previous work, publications, and other sources. This form of data collecting is chosen because it allows for comparison with previous research. This includes the internet, which is a data-collecting tool. The internet was utilized to find information about tough or unclear subjects.

**3.3 System Modeling**

A system model is a conceptual model resulting from system modeling that describes and represents a system. It is an interaction between a set of components that work together to achieve a common purpose. Visual models of object-oriented software-intensive systems may be created utilizing a set of visual notation techniques included in the Unified Modeling Language, which is used in the development of this modern system. UML diagrams utilized in this new design include use case diagrams, class diagrams, and activity diagrams.

**3.3.1 Use Case Diagrams**

Use cases are collections of interactions between systems and users. Use case diagrams are used to visually summarize a system's functionality in terms of its actors, its goals (represented as use cases), and any dependencies between those use cases

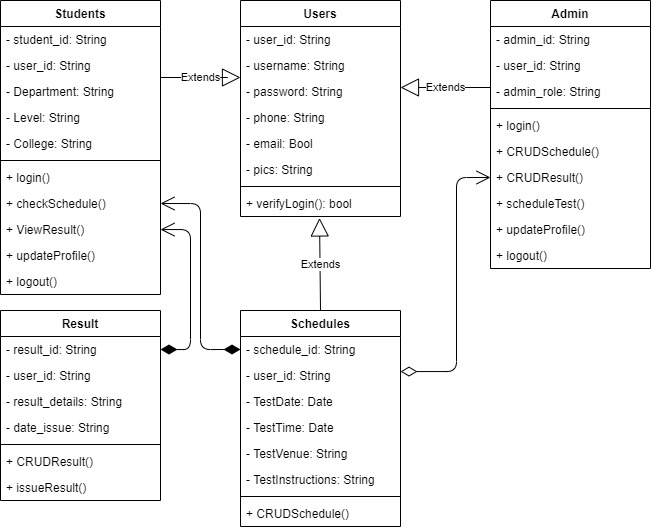


**Fig 3.1 System Use Case Diagram**

**]**

**3.3.2 Class Diagrams**

The Unified Modeling Language (UML) class diagram is an implementation of an independent view of how the system interface will be, with each class having its own properties and illustrating how they interact with one another. Class diagrams use the rules established by the Unified Modeling Language to visually depict the static structure and composition of a specific system (UML).

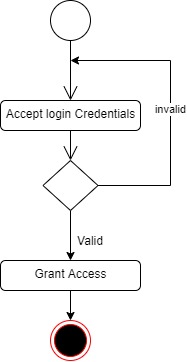


**Fig 3.2 System Class Diagram**

**3.3.3 Activity Diagrams**

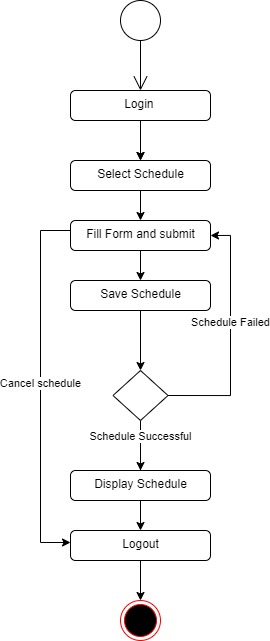
Similar to a flowchart or a data flow diagram, an activity diagram visually depicts a sequence of events or the flow of control in a system, but it functions more like an advanced version of both.

**Login**

The process for gaining access to the system is depicted in the diagram below; the username and password must be accurate to gain access.

**Fig 3.3 System Login Activity Diagram**

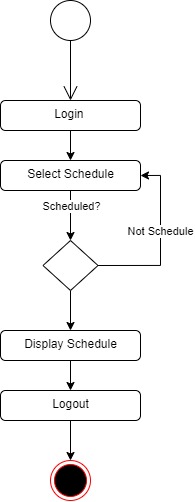
**Create Schedule**

The process for automatically scheduling student medical test dates is depicted in the diagram below; The system ensures that the user is authenticated and authorized to perform the scheduling.

**Fig 3.4 Create Schedule Activity Diagram**

**View Schedule**

The process for viewing medical test dates by the student is depicted in the diagram below; The system ensures that the user is authenticated and authorized to view the scheduled date for validity purposes.



**Fig 3.5 View Schedule Activity Diagram**

**3.4 Database Design**

The logical explanation of how data is kept in the computer's memory is called input specification. The freedom experienced in using the system and the convenience of retrieving and reading the data and assuring applicability across the internet make SQL standards essential for ensuring that structured data is uniform and independent of applications.

The following are some of the input specifications used in this project work.

1. Users Table: contains the generic information of all system users.
2. Students Table: contains every system-registered student information.
3. Schedules Table: contains every system information regarding schedules

**Table 3.1 Users Input Specification Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Null** | **Key** | **Length** | **Description** |
| user\_id | Varchar | No | PK | 32 | Unique string for identifying users |
| username | Varchar | No |  | 20 | Unique name for users |
| password | Varchar | No |  | 128 | User Password |
| phone | Varchar | No |  | 11 | User phone number |
| email | Varchar | No |  | 100 | User email address |
| pics | Varchar | No |  | 100 | User profile picture |

**Table 3.2 Students Input Specification Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Null** | **Key** | **Length** | **Description** |
| student\_id | Varchar | No | PK | 32 | Unique string for identifying citizens |
| user\_id | BigInt | No |  | 10 | Reference to the User table |
| Department | BigInt | No |  | 60 | Citizen full name |
| Level | Varchar | No |  | 60 | Citizen father name |
| College | Varchar | No |  | 60 | Citizen mother name |

**Table 3.3 Schedules Input Specification Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Null** | **Key** | **Length** | **Description** |
| schedule\_id | Varchar | No | PK | 32 | Unique string for identifying citizens |
| user\_id | BigInt | No |  | 10 | Reference to the User table |
| TestDate | BigInt | No |  | 60 | Citizen full name |
| TestTime | Varchar | No |  | 60 | Citizen father name |
| TestVenue | Varchar | No |  | 60 | Citizen mother name |
| TestInstructions | Date | No |  | 20 | Citizen’s Date of birth |

**3.5 Output Design**

This declares and displays the outcome of the given input. This system's output is dependent on its input. The output specification is listed below.

**Table 3.4 Users** **output design table**

**MED-TEST SCHEDULER AND ISSUANCE SYSTEM**

**List of the System Registered Users**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User\_id** | **Username** | **Phone** | **Email** | **Is\_admin** |
| XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX |

**Table 3.5 Students** **output design table**

**MED-TEST SCHEDULER AND ISSUANCE SYSTEM**

**List of the System Registered Students**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **student\_id** | **user\_id** | **Department** | **Level** | **College** |
| XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX |

**Table 3.6 Schedule** **output design table**

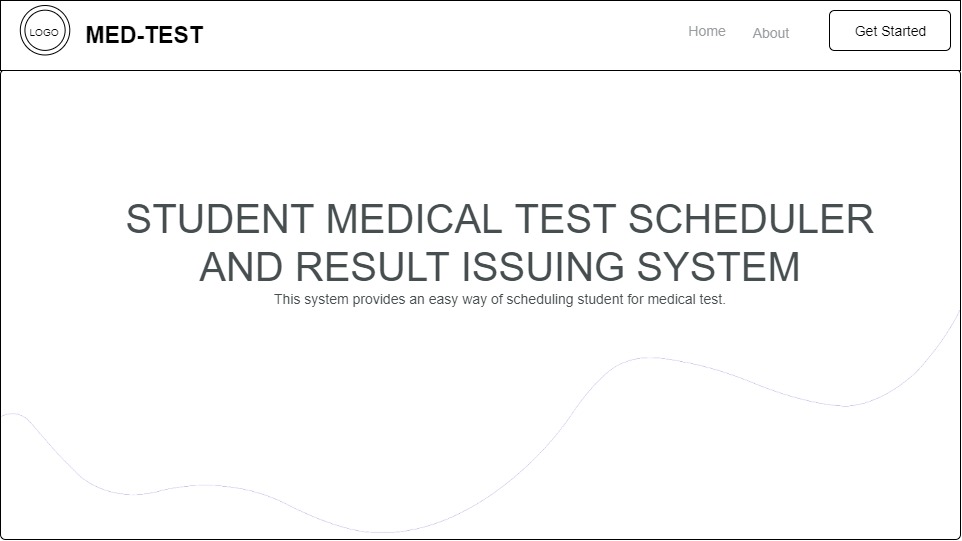
**MED-TEST SCHEDULER AND ISSUANCE SYSTEM**

**List of the System Saved Schedule**

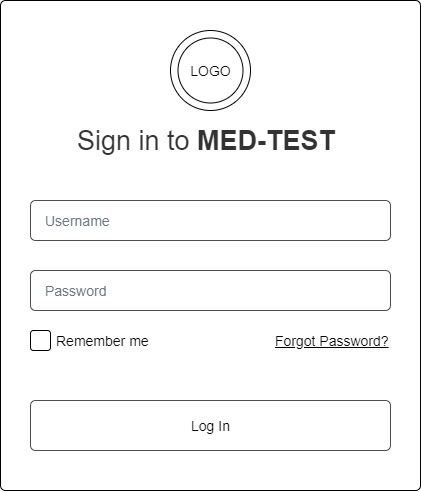
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **schedule\_id** | **user\_id** | **TestDate** | **TestVenue** | **TestInstructions** | **TestTime** |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXX |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXX |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXX |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXX |

**3.6 Input & User Interface Design**

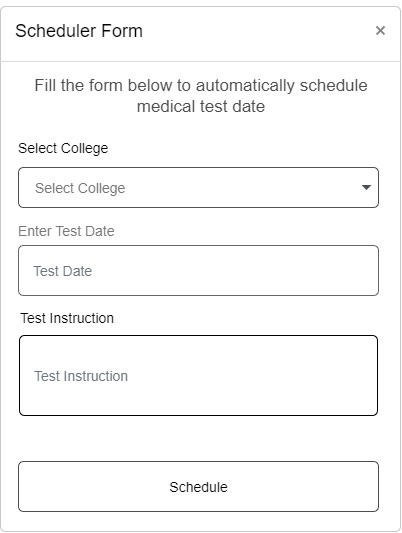
This is a graphic depiction of the system interface; it will be designed to be user-friendly, responsive, and visually beautiful. Furthermore, it will be appropriately safeguarded, thus authorization will be required to see certain levels of the information. To help with the designs, a mid-fidelity wireframing program called Draw.io is employed.



**Fig 3.6.1 Home Page**



**Fig 3.7 Login Form**



**Fig 3.8 Scheduler Form**

**3.7 System Requirement**

Every software system built has a stated system requirement on which it is meant to execute for the best performance. The system requirements, on the other hand, are the bare minimum of hardware and software required for the system to work properly.

**3.7.1 Hardware Requirement**

System Hardware Requirement Include:

1. Minimum of 2 GB of RAM (Random Access Memory).
2. Minimum of Intel Dual core processor.
3. Minimum of 250GB HDD (Hard Disk Drive).

**3.7.2 Software Requirement**

The software requirements include:

1. At least windows 7 OS (Operating System).
2. Vs. Code IDE installation.
3. Browsers include Chrome and Firefox.

**3.8 Choice of Programming Language**

This study will be a web-based application based on a relational database (SQLite) design. HTML (hypertext markup language), CSS (cascading style sheet), and JavaScript will be utilized for front-end development, while Django (Python) will be used for back-end programming.

**CHAPTER FOUR**

# SYSTEM IMPLEMENTATION EVALUATION

# 4.1 Introduction

This section covers in detail how the new system is implemented to ensure its effectiveness. It depicts functional (new) system examples as well as how the system is to be implemented.

# System Testing and Evaluation

There are various reasons why the developed system should be tested. For example, only via testing will we be able to identify any flaws in the new system and provide solutions. This project employed both unit and integration testing to validate the design's efficacy and efficiency, as well as to guarantee the new system meets its functional requirements and is error-free.

**Unit Testing**

This part examines specific units or single components of the system individually to confirm that specific phases function properly and without problems.

**Integration Testing**

The software was tested utilizing integration testing, in which all parts were assembled and functioned as one. The connectivity between the various components was checked to ensure they are properly integrated and the units can work together.

# 4.3 System Installation

In order to use the proposed application on any computer system, the following steps need to be taken:

1. Make sure, pip, pipenv, and python3 or greater are installed on the system.
2. Copy your project folder to any location of your choice.
3. Open project folder in Visual Studio Code
4. On the terminal run “pipenv install -r requirements.txt”
5. On the terminal run “python manage.py runserver”
6. Open any browser on the system example Chrome, Microsoft Edge, Mozilla Firefox.
7. On the address bar, type <http://127.0.0.1> and press the enter key the site should be loaded.

# 4.4 Security Measures

Since the scope of the application is public, literally all the information is made available to any user, but some functionalities are restricted to the hospital admin and functionalities that have to do with updating the hospital profile, creating hospital staff, handling birth registrations, etc. The restriction is carried out by using passwords when the hospital admin webpage is accessed. Notwithstanding some functionalities are also restricted to the specific type of user.

# 4.6 Sample Outputs

These describe and give the pictorial representation of the program or software; it shows and gives a clear understanding of the design, and displays all the interfaces.

**4.6.1 Homepage**

The picture below represents the homepage, which is the website's initial page. It is the initial page that a person sees when they visit a website's domain or click on a link to it. This page's objective is to offer an overview of the website's content and features.

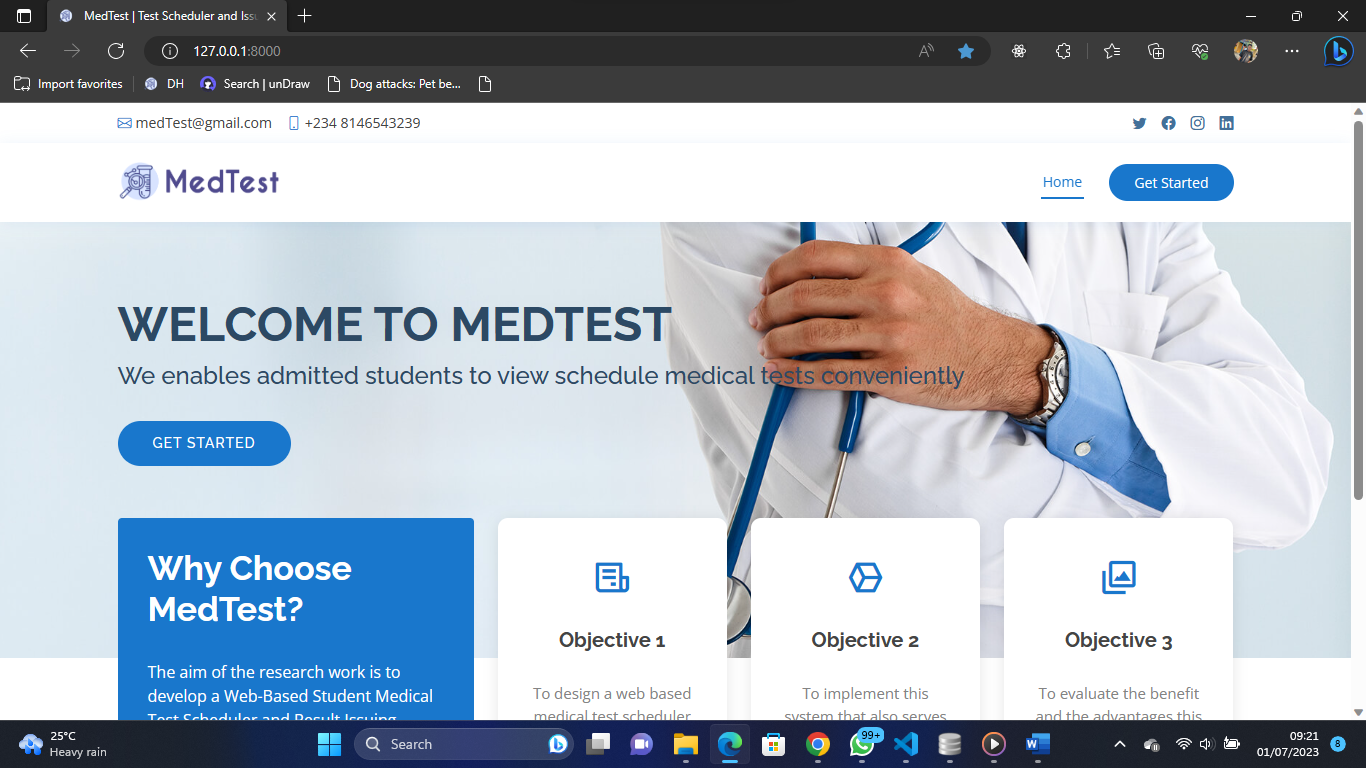


Fig 4.6.1: Homepage

**4.6.2 User Login**

This is a page that grants users (admin, and staff) access to the system only if the correct credentials are provided.

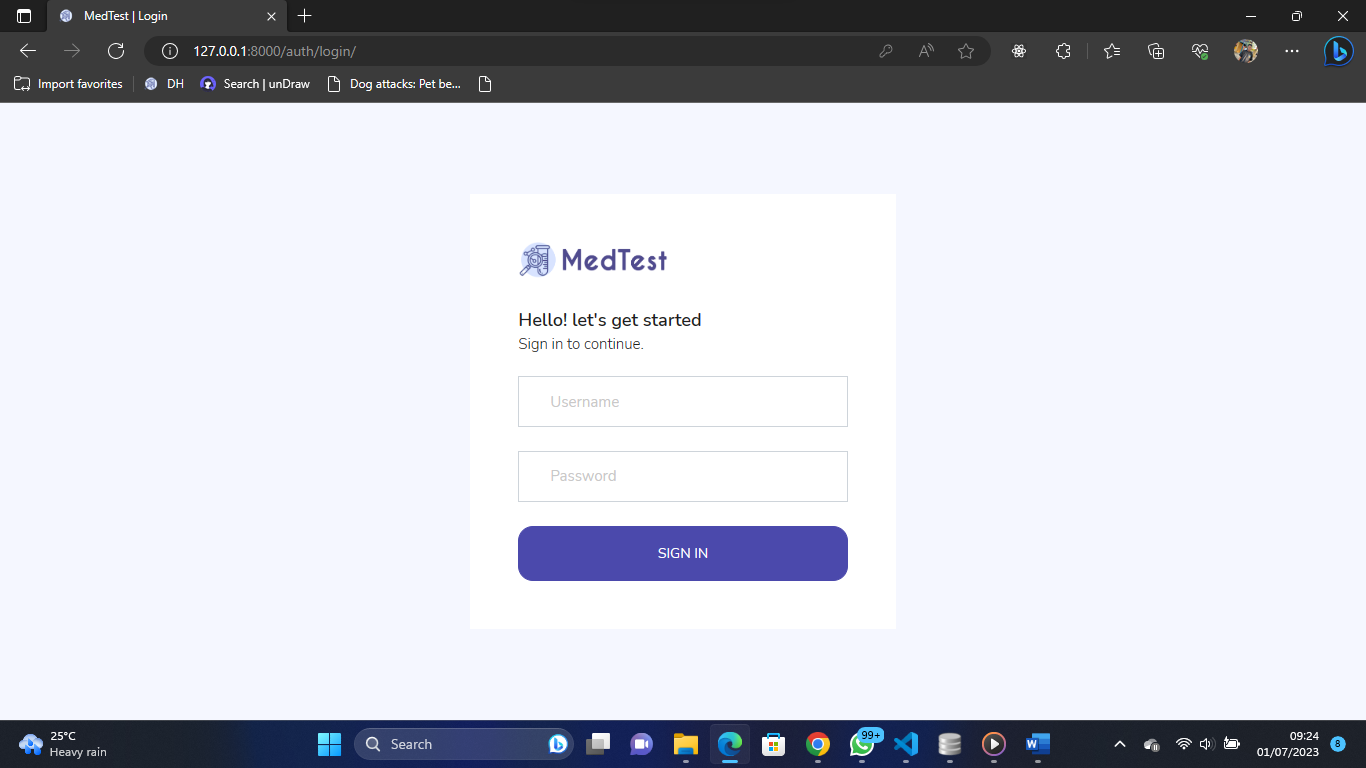


Fig 4.6.2: User Login

**4.6.3 Hospital Admin Dashboard**

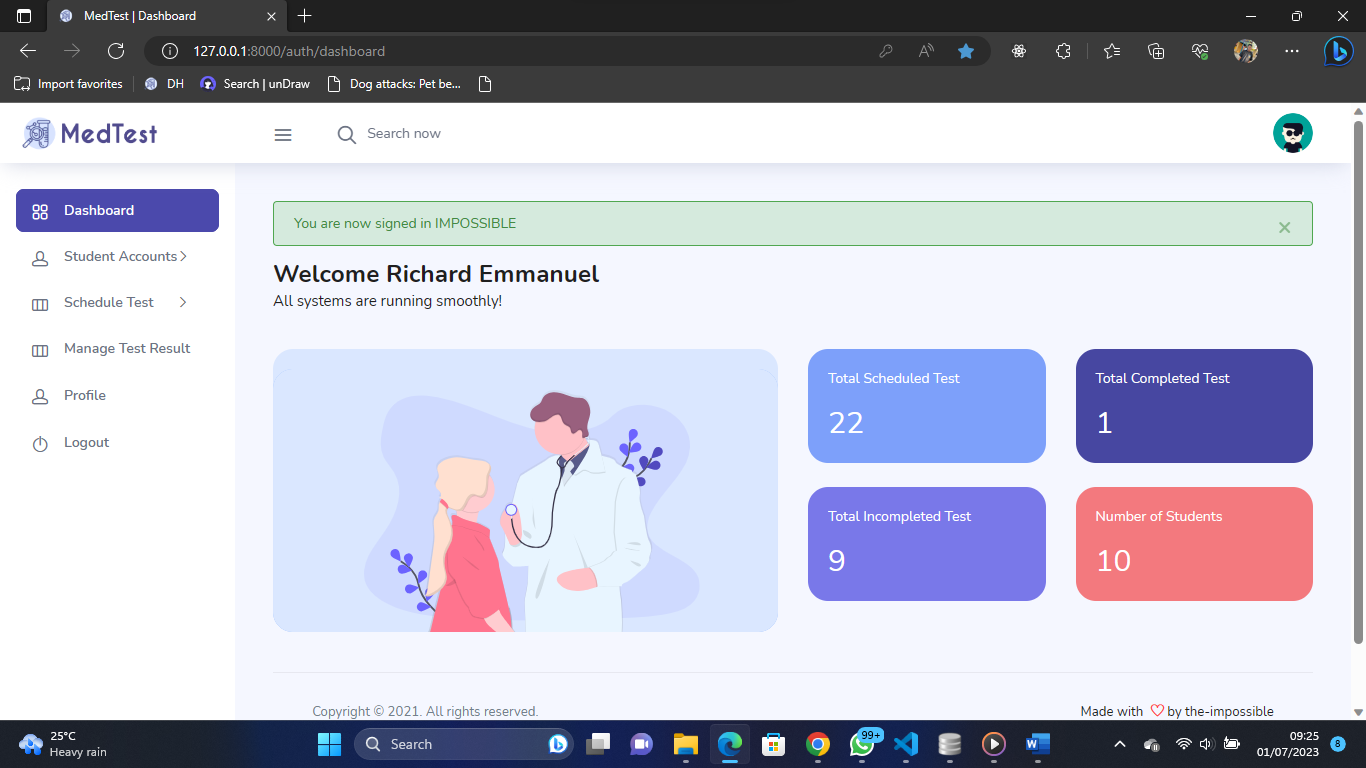
This is the hospital admin dashboard, the sidebar shows the available functionality for the admin

Fig 4.6.3: Hospital Admin Dashboard

**4.6.4 Create a Student Account**

This is the page where the hospital admin can create an account for the student individually

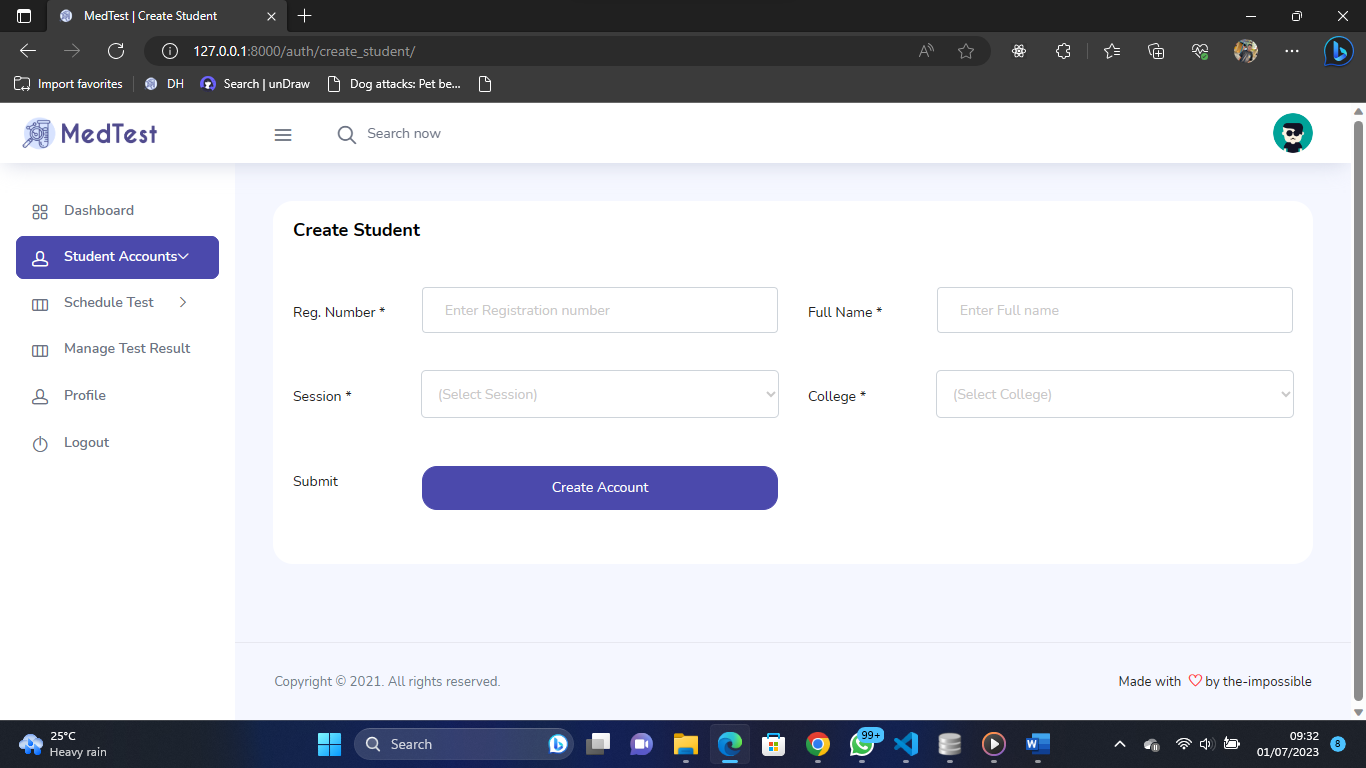


Fig 4.6.4: Create a Student Account

**4.6.5 Batch Create Student’s Account**

This is the page where the hospital admin can create multiple student accounts at a go.

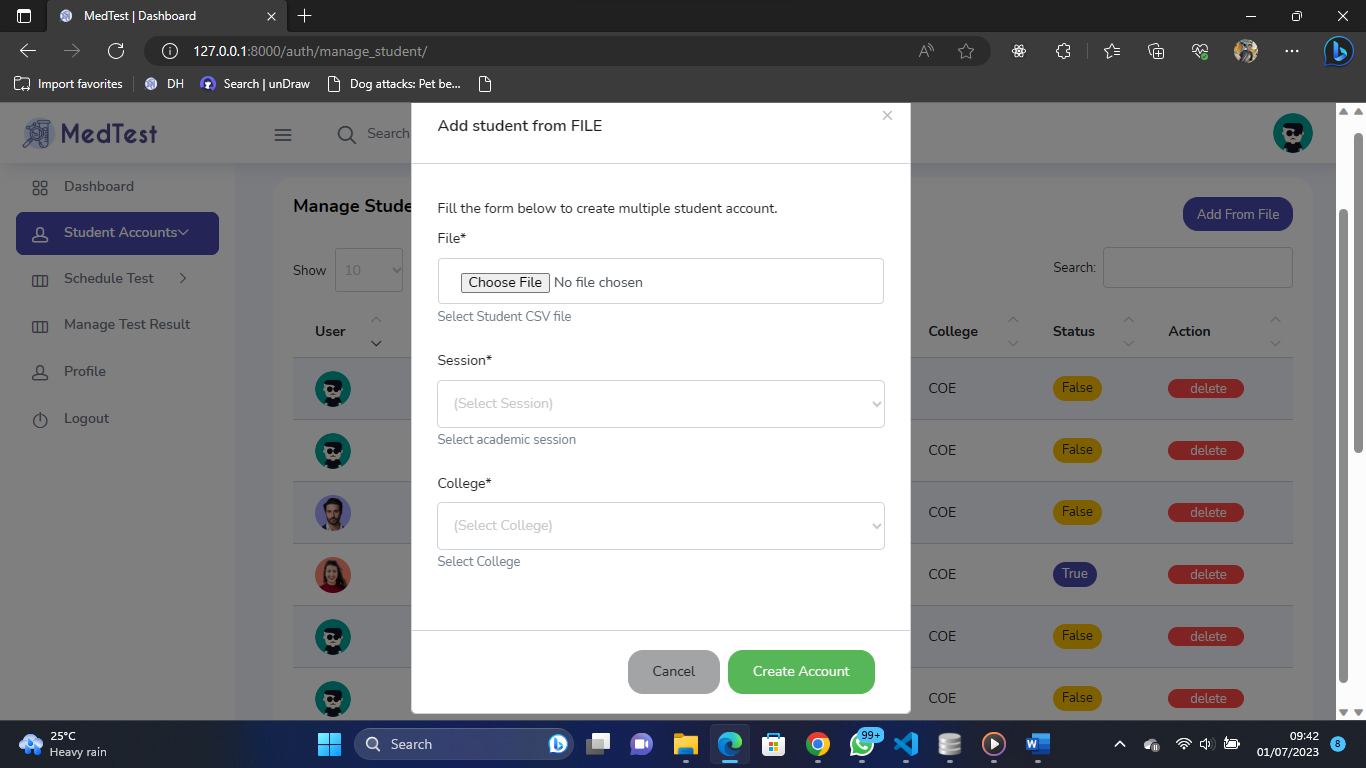


Fig 4.6.5: Batch Create Student’s Account

**4.6.6 Manage Student’s Account**

This is the page where the hospital admin can effectively manage each student account

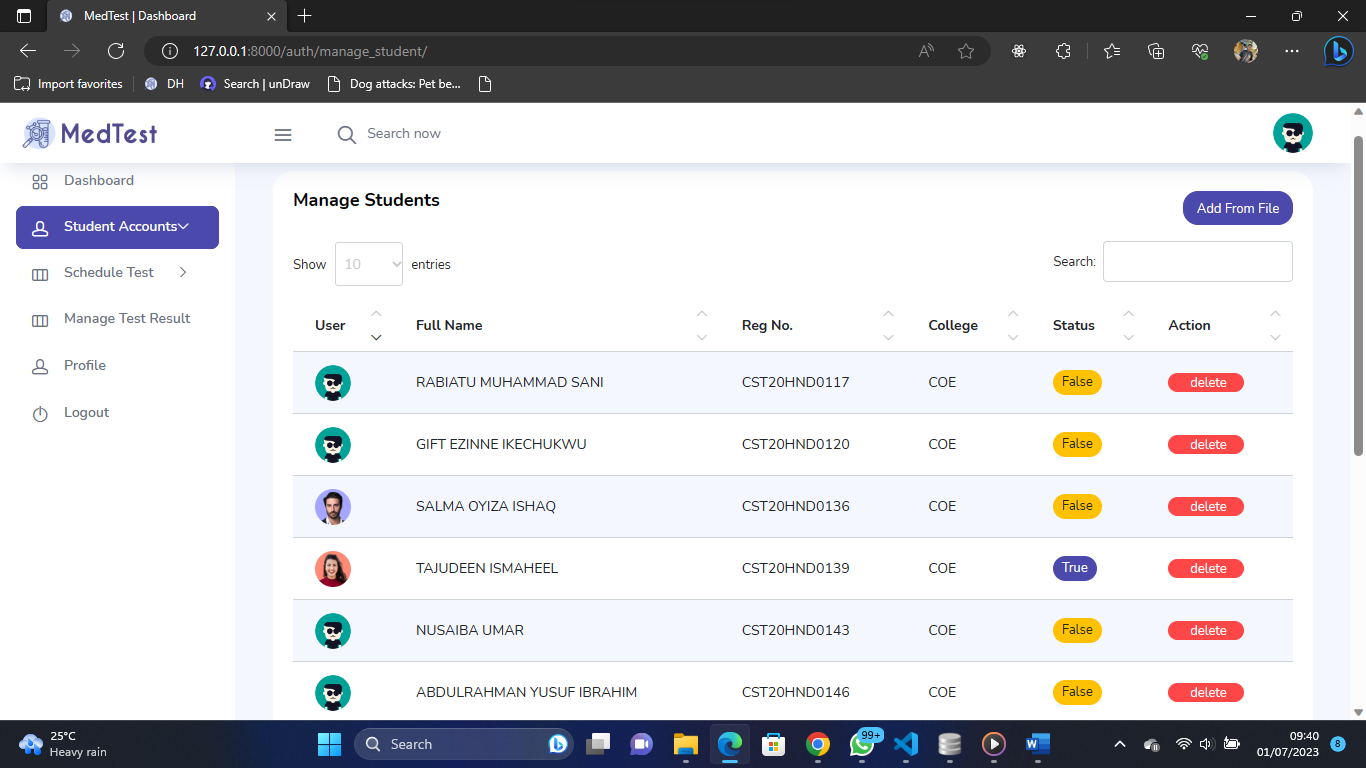


Fig 4.6.6: Manage Student’s Account

**4.6.7 Schedule Test**

The page form on submit is used to carry out the automatic scheduling of tests for each student

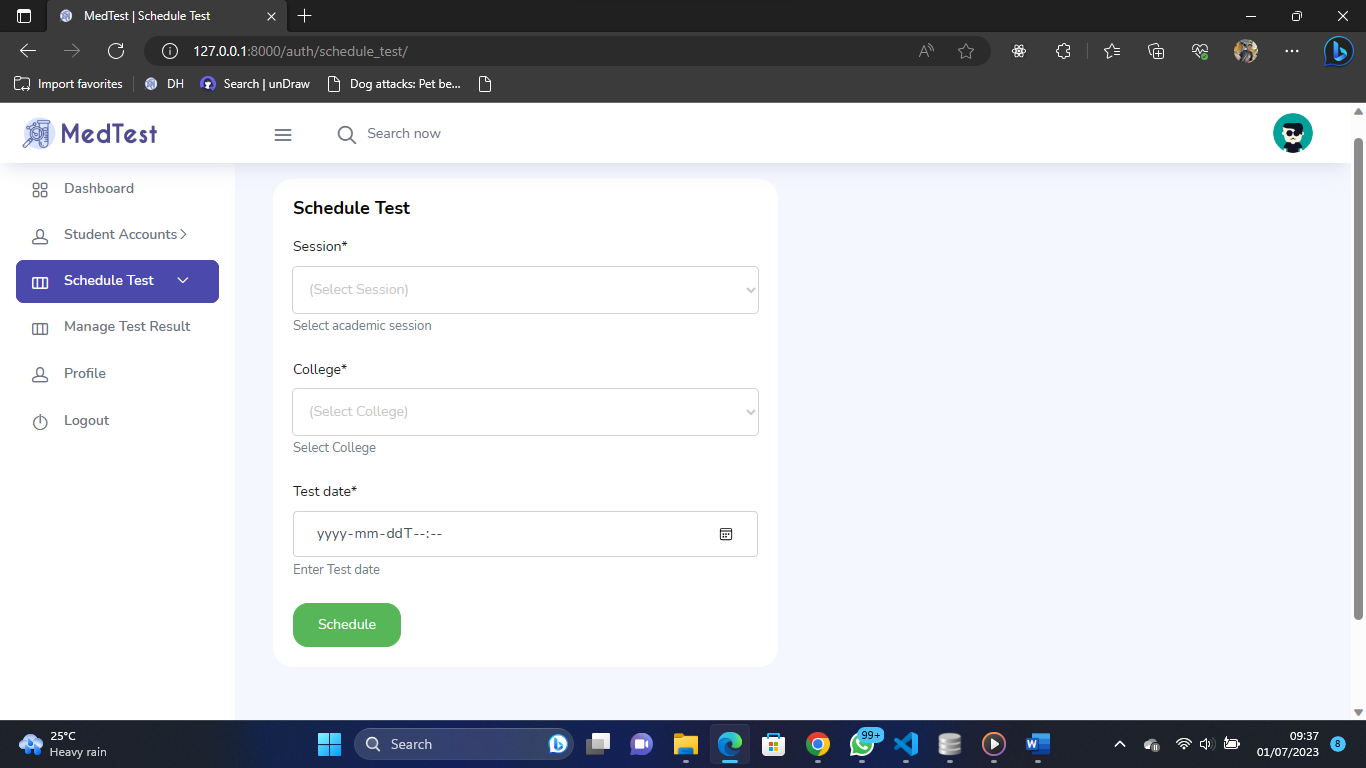


Fig 4.6.7: Schedule Test

**4.6.8 Manage Test Result**

Admin can use this page to make changes to each student test result effectively

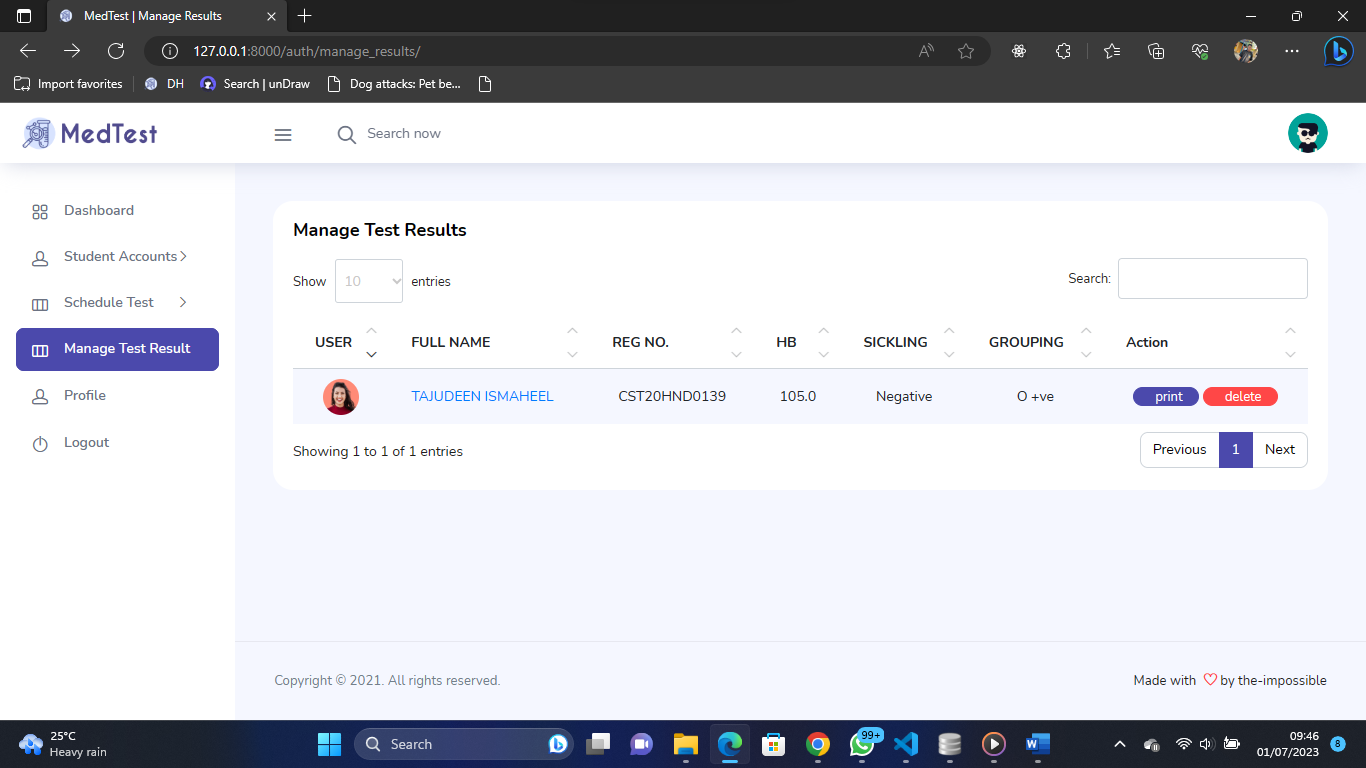


Fig 4.6.8: Manage Test Result

**4.6.9 Update Student Test Result**

This page enables admin to upload test result for the students that has participated in the test

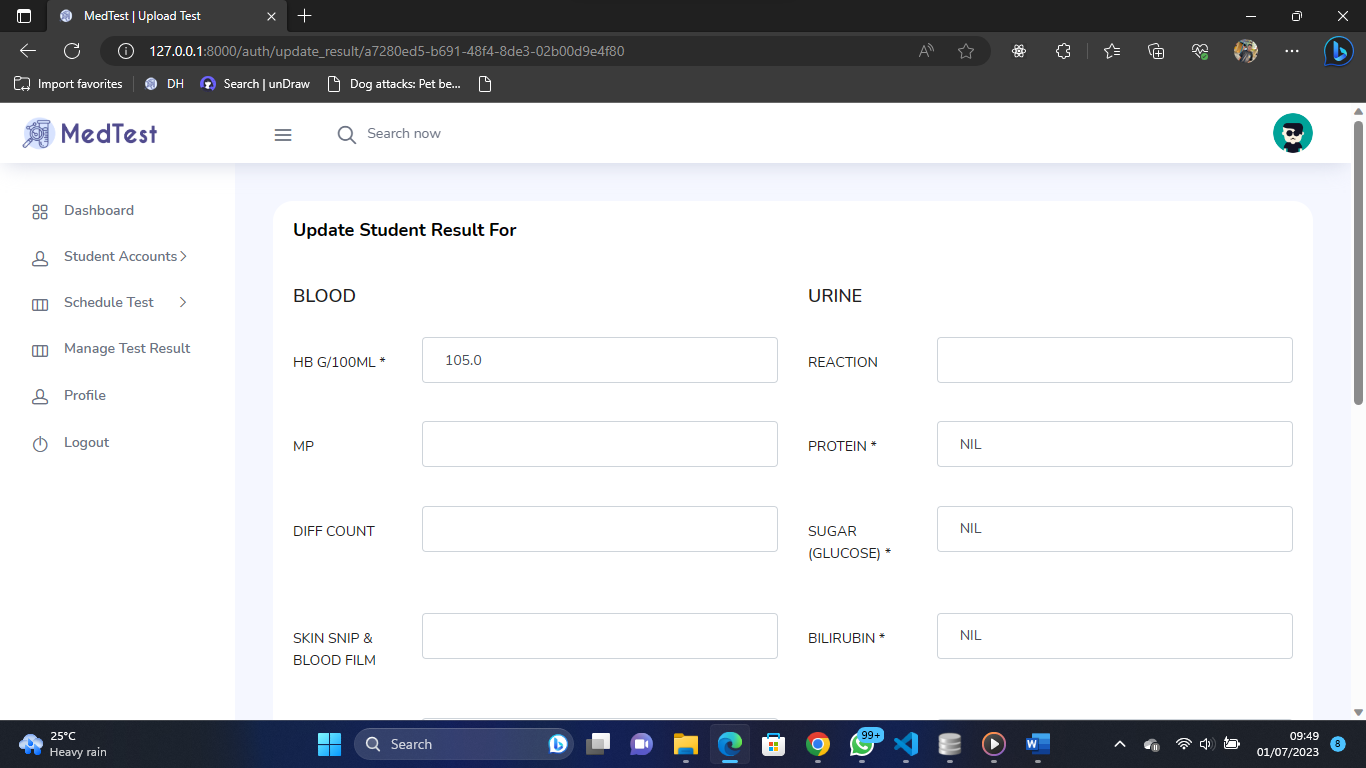


Fig 4.6.9: Update Student Test Result

**4.6.9.1 Account Profile**

This page performs updates on the profile of every authenticated user

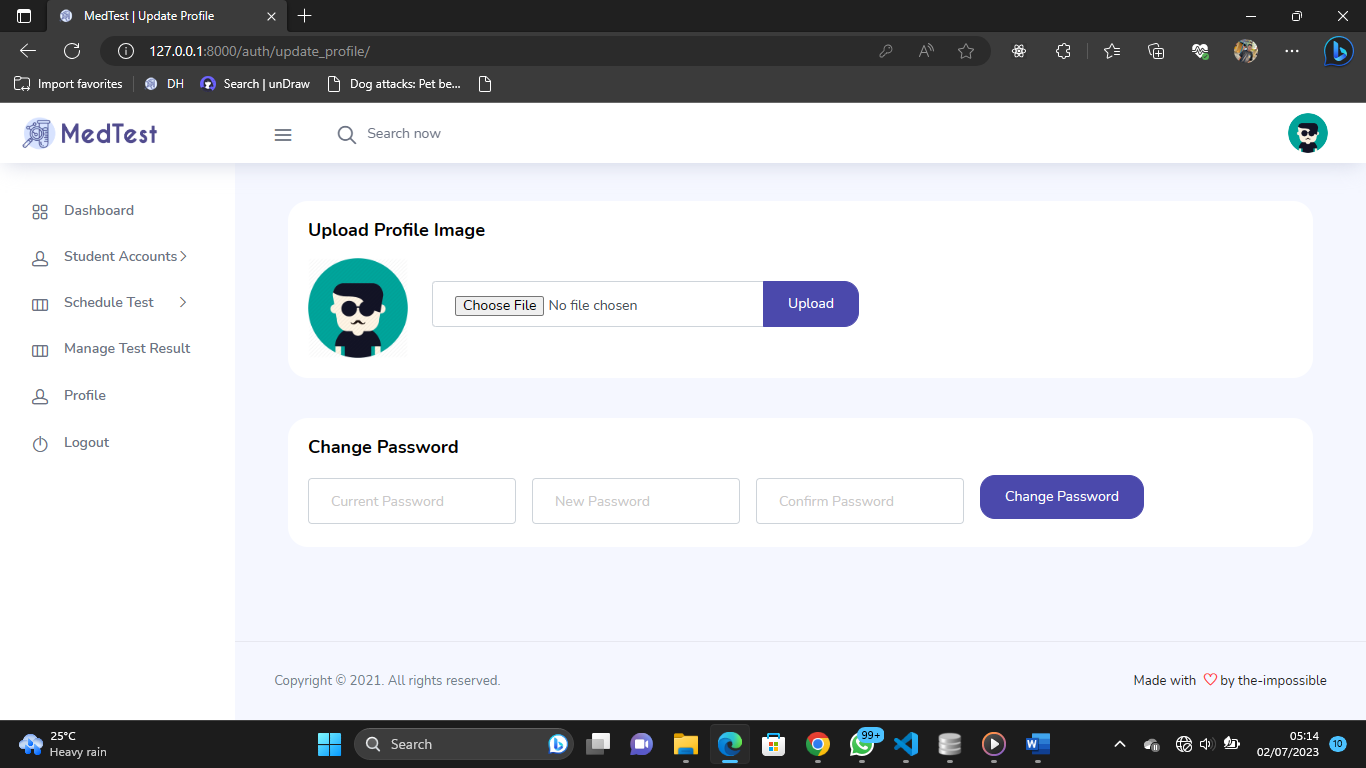


Fig 4.6.9.1: Account Profile

**4.6.9.2 Student Dashboard**

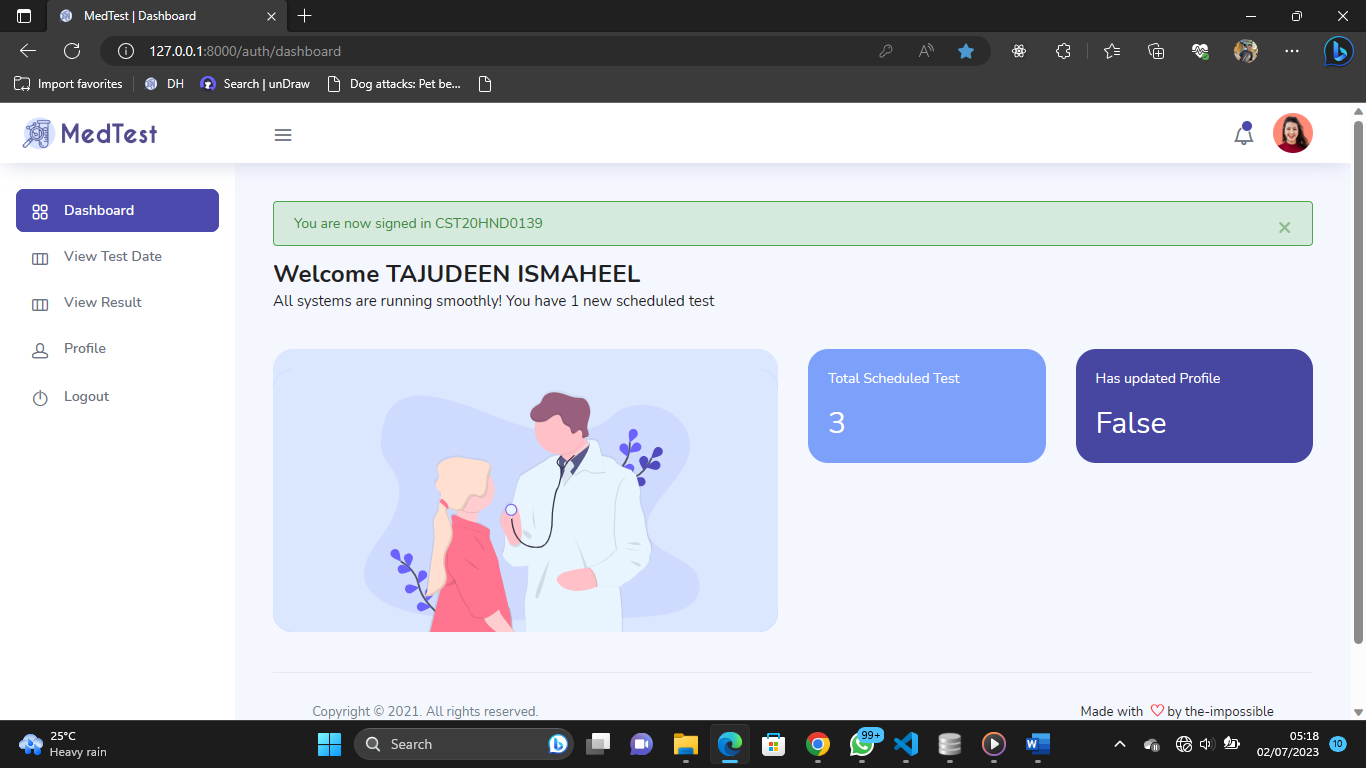
This is the student dashboard, the sidebar shows the available functionality for the student 

Fig 4.6.9.2: Student Dashboard

**4.6.9.3 Student Scheduled Test**

Student can use this page to view their scheduled date for the test

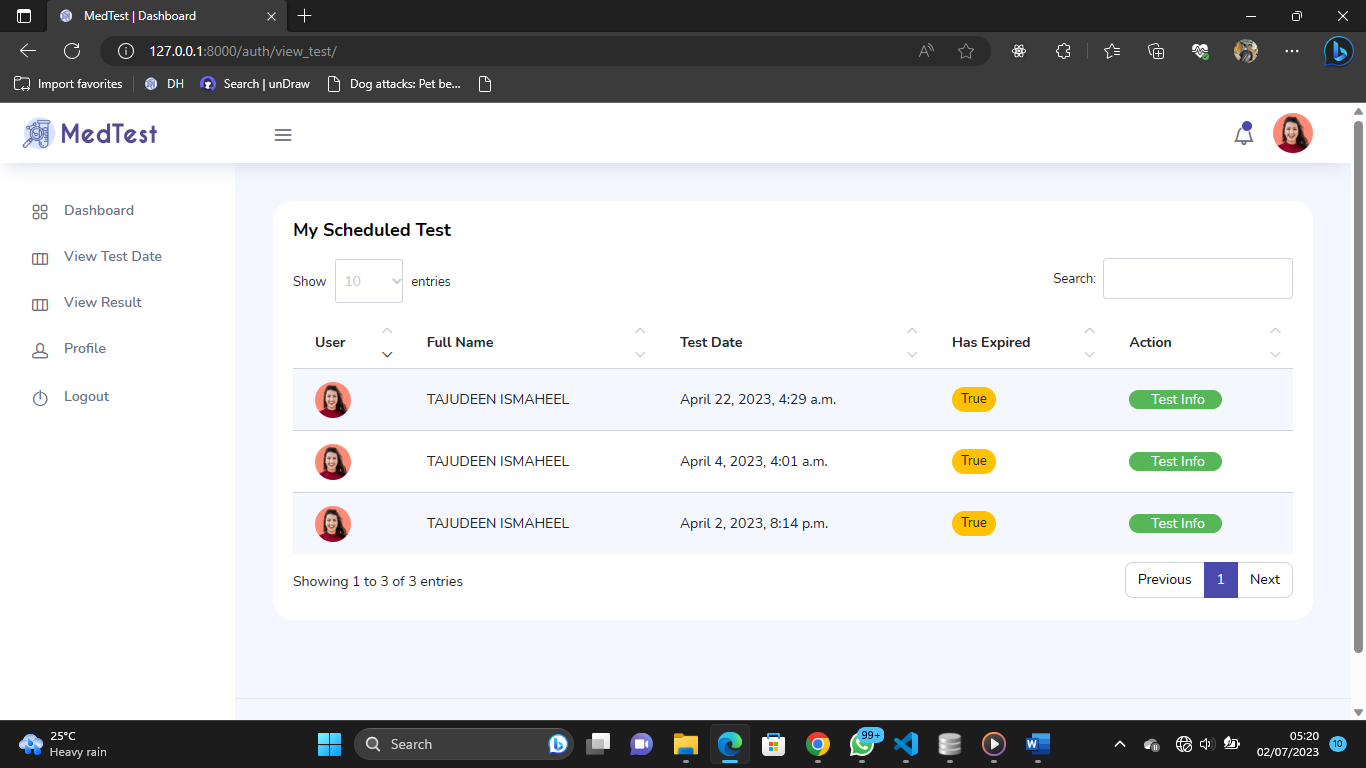


Fig 4.6.9.3: Student Scheduled Test

**4.6.9.4 View Test Result**

If test is completed and the result uploaded, it can be viewed and downloaded by the student

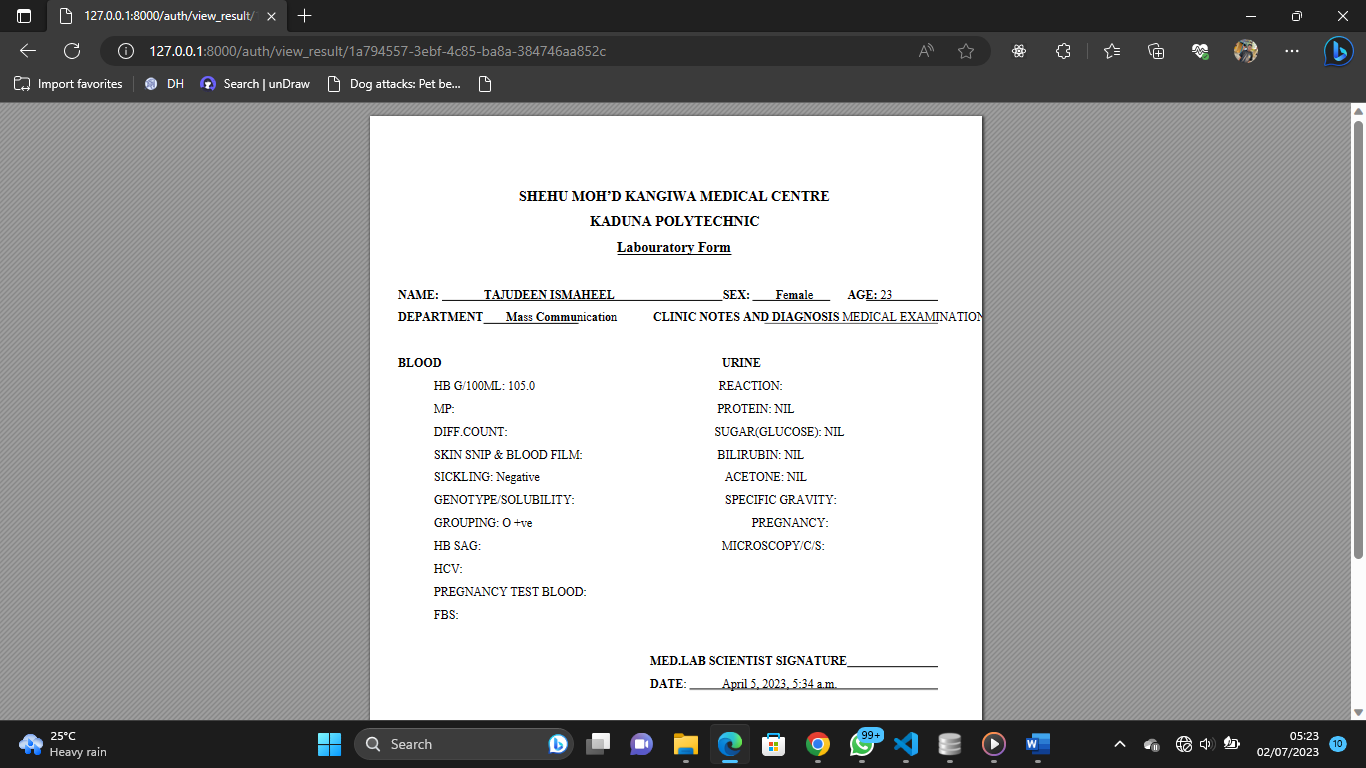
****

Fig 4.6.9.4: View Test Result

**CHAPTER FIVE**

# SUMMARY CONCLUSION AND RECOMMENDATION

# 5.1 Summary

The research work focuses on developing a web-based student medical test scheduler and result issuing system for Nigerian Polytechnic Institutions. The traditional manual process of scheduling and conducting medical tests for newly admitted students has proven to be time-consuming and lacking in organization. The aim of the system is to eliminate student waiting time during registration by providing an efficient and convenient method for scheduling and issuing medical test results. Through a user-friendly web interface, students can schedule their tests online and access their results, leading to improved efficiency, reduced waiting times, and enhanced accessibility for both students and medical staff.

**5.2 Conclusion**

In conclusion, the development of a web-based student medical test scheduler and result issuing system offers a transformative solution to the challenges faced in the traditional manual process. By leveraging technology, the system streamlines the scheduling process, eliminates long queues and waiting times, and provides students with convenient online access to their medical test results. The implementation of this system not only enhances efficiency and organization but also improves the overall experience for both students and medical staff. With its user-friendly interface and reliable database, the system revolutionizes the way medical tests are conducted and managed in Nigerian Polytechnic Institutions, paving the way for a more seamless and effective registration process.

# 5.2 Recommendation

Based on the research findings, the following recommendations are suggested:

1. Adoption of Web-Based System: Nigerian Polytechnic Institutions should adopt the web-based student medical test scheduler and result issuing system as it will streamline the registration process and reduce waiting time for students.
2. Training and Orientation: Adequate training and orientation programs should be conducted for both students and medical staff to ensure they have a proper understanding of the web-based system. Training sessions will help users navigate the system effectively, reducing any confusion or errors during its usage.
3. Regular Maintenance and Updates: Regular maintenance and updates of the web-based system are essential to keep it running smoothly. Technical issues, if any, should be promptly addressed, and updates should be applied to improve functionality and security.
4. Integration with Student Information Systems: The web-based system should be integrated with existing student information management systems. This integration will facilitate seamless data synchronization, avoiding the need for duplicate data entry and ensuring accurate and up-to-date information.
5. Data Security and Privacy: Measures should be taken to ensure the security and privacy of student medical information within the web-based system. Robust data encryption and access control mechanisms should be implemented to protect sensitive data from unauthorized access.

By implementing these recommendations, Nigerian Polytechnic Institutions can enhance their medical test scheduling and result issuing process, leading to improved efficiency, better student experience, and a well-organized medical database for the institution.

**REFERENCES**

Cara R., O. (2017). Online Medical Appointment Scheduling System: *Higher Technical School*

*of Computer Engineering Polytechnic University of Valencia*

Ele, B., Odey, A. J., Frank, E. N., & Ekinya, M. I. (2020). A Web-Based Medical Appointment

Scheduling with SMS Alert Notification System. *Transactions on Machine Learning and Artificial Intelligence, 8*(6), 28–38. <https://doi.org/10.14738/tmlai.86.9098>

Hema Kumar, S., Uday Kiran, J., Ambeth Kumar, V. D., Saranya, G., & Ramalakshmi, V.

(2019). Effective online medical appointment system. International Journal of Scientific and Technology Research, 8(9), 803–805.

Noorsyahira, I., Shahreen K., Yusmadi Y. J., Rohayanti H., Ayu A. (2017). Medical

Appointment Application. *Creative Commons Attribution License* <https://doi.org/10.26480/aem.02.2017.05.09>

Sanjeevani P. A., Wrushali R. A., Pallavi A. C., Puja T. C., Bhil, N. K. (2018). Doctor

Appointment Online Booking System: *The International Journal of Creative Research Thoughts (IJCRT) 6*(2). https://ijcrt.org/papers/IJCRT1812133.pdf

**APPENDIX**

**Views.py**

from django.shortcuts import render, redirect

from django.contrib.auth.mixins import LoginRequiredMixin

from django.http import HttpResponseRedirect

from django.contrib.auth import authenticate, login, logout

from django.views.generic import ListView, DetailView, CreateView, UpdateView, DeleteView, TemplateView

from django.contrib.messages.views import SuccessMessageMixin

from django.contrib.auth.hashers import make\_password, check\_password

from django.contrib import messages

from django.views import View

import csv, io, codecs

from django.urls import reverse\_lazy

from MedTest\_auth.models import \*

from MedTest\_auth.forms import \*

PASSWORD = '12345678'

# Create your views here.

class DashboardView(TemplateView):

    template\_name = "auth/dashboard.html"

class LoginView(View):

    def get(self, request):

        return render(request, 'auth/login.html')

    def post(self, request):

        username = request.POST.get('username').upper().strip()

        password = request.POST.get('password').strip()

        if username and password:

            user = authenticate(request, username=username, password=password)

            if user:

                if user.is\_active:

                    login(request, user)

                    messages.success(request, f"You are now signed in {user}")

                    nxt  = request.GET.get('next', None)

                    if nxt is None:

                        return redirect('auth:dashboard')

                    return redirect(self.request.GET.get('next', None))

                else:

                    messages.warning(request, 'Account not active contact the administrator')

            else:

                messages.error(request, 'Invalid login credentials')

        else:

            messages.error(request, 'All fields are required!!')

        return redirect('auth:login')

class LogoutView(LoginRequiredMixin, View):

    def post(self, request):

        logout(request)

        messages.success(request, 'You are successfully logged out, to continue login again')

        return redirect('auth:login')

class ManageStudentAccount(ListView):

    template\_name = 'auth/manage\_student.html'

    def get\_queryset(self):

        return StudentProfile.objects.all()

    def get\_context\_data(self, \*\*kwargs):

        context = super().get\_context\_data(\*\*kwargs)

        context["form"] = CreateStudentForm

        return context

    def post(self, request):

        form = CreateStudentForm(request.POST, request.FILES)

        if form.is\_valid():

            csv\_obj = csv.reader(codecs.iterdecode(request.FILES['file'], 'utf-8'))

            objs = []

            sub\_objs = []

            session = form.cleaned\_data.get('session')

            college = form.cleaned\_data.get('college')

            for row in csv\_obj:

                objs.append(User(username=row[0], name=row[1], password=make\_password(PASSWORD)))

            created\_users = User.objects.bulk\_create(objs)

            for user in created\_users:

                sub\_objs.append(StudentProfile(user\_id=user, session=session, college=college))

            created\_user\_profiles = StudentProfile.objects.bulk\_create(sub\_objs)

        else:

            messages.error(self.request, form.errors.as\_text())

            return render(request, 'auth/manage\_student.html',

            context={

                'form':form,

                'object\_list':self.get\_queryset()

            })

        return HttpResponseRedirect(self.get\_success\_url())

    def get\_success\_url(self):

        return reverse("auth:manage\_student")

class StudentDeleteView(SuccessMessageMixin, DeleteView):

    model = User

    success\_message = "Student has been deleted!"

    def get\_success\_url(self):

        return reverse("auth:manage\_student")

class StudentAccountView(CreateView):

    model = User

    template\_name = "auth/create\_student.html"

    form\_class = SingleCreateStudentForm

    def get\_success\_url(self):

        return reverse("auth:manage\_student")

    def form\_valid(self, form):

        college = form.cleaned\_data.get('college')

        session = form.cleaned\_data.get('session')

        form.instance.password = make\_password(PASSWORD)

        form = super().form\_valid(form)

        messages.success(self.request, f"Account created for {self.object.username}")

        StudentProfile.objects.create(user\_id=self.object, session=session, college=college)

        return form

class ManageTest(ListView):

    template\_name = 'auth/manage\_test.html'

    def get\_queryset(self):

        return ScheduleTest.objects.filter(result\_uploaded=False).order\_by('-test\_date')

    def get\_success\_url(self):

        return reverse("auth:manage\_test")

    def get\_context\_data(self, \*\*kwargs):

        context = super().get\_context\_data(\*\*kwargs)

        context["form"] = ScheduleTestForm

        return context

    def post(self, request):

**Homepage**

{% extends 'base.html' %}

{% load static %}

{% block title %}Test Scheduler and Issuance System{% endblock title %}

{% block head %}{% include 'partials/head.html' %}{% endblock head %}

{% block body %}

  {% block nav %} {% include 'partials/nav.html' %} {% endblock %}

  <!-- ======= Hero Section ======= -->

  <section id="hero" class="d-flex align-items-center">

    <div class="container">

      <h1>Welcome to MedTest</h1>

      <h2>We enables admitted students to view schedule medical tests conveniently</h2>

      <a href="{% url 'auth:login' %}" class="btn-get-started scrollto">Get Started</a>

    </div>

  </section><!-- End Hero -->

  <main id="main">

    <!-- ======= Why Us Section ======= -->

    <section id="why-us" class="why-us">

      <div class="container">

        <div class="row">

          <div class="col-lg-4 d-flex align-items-stretch">

            <div class="content">

              <h3>Why Choose MedTest?</h3>

              <p>

                The aim of the research work is to develop a Web-Based Student Medical Test Scheduler and Result Issuing System to eliminate student waiting time during registration.

              </p>

              <div class="text-center">

                <a href="#" class="more-btn">Get Started <i class="bx bx-chevron-right"></i></a>

              </div>

            </div>

          </div>

          <div class="col-lg-8 d-flex align-items-stretch">

            <div class="icon-boxes d-flex flex-column justify-content-center">

              <div class="row">

                <div class="col-xl-4 d-flex align-items-stretch">

                  <div class="icon-box mt-4 mt-xl-0">

                    <i class="bx bx-receipt"></i>

                    <h4>Objective 1</h4>

                    <p>To design a web based medical test scheduler that enables admitted students schedule online for their medical tests conveniently</p>

                  </div>

                </div>

                <div class="col-xl-4 d-flex align-items-stretch">

                  <div class="icon-box mt-4 mt-xl-0">

                    <i class="bx bx-cube-alt"></i>

                    <h4>Objective 2</h4>

                    <p>To implement this system that also serves as a result issuer</p>

                  </div>

                </div>

                <div class="col-xl-4 d-flex align-items-stretch">

                  <div class="icon-box mt-4 mt-xl-0">

                    <i class="bx bx-images"></i>

                    <h4>Objective 3</h4>

                    <p>To evaluate the benefit and the advantages this system will have over the pre-existing traditional way of scheduling these medical tests</p>

                  </div>

                </div>

              </div>

            </div><!-- End .content-->

          </div>

        </div>

      </div>

    </section><!-- End Why Us Section -->

  </main><!-- End #main -->

  {% block footer %}{% include 'partials/footer.html' %}{% endblock footer %}

  <div id="preloader"></div>

  <a href="#" class="back-to-top d-flex align-items-center justify-content-center"><i class="bi bi-arrow-up-short"></i></a>

{% block script %}{% include 'partials/script.html' %}{% endblock script %}

{% endblock body %}