BIOM9540 – Major Project Report

Patient Management and Diagnostic Reporting System

November 30th, 2024

Team Members – DIAGNOSYS

Team Member	zID
Alice Gao	z5359533
Vinoshan Parathan	z5358838
Rohit Benjamin George	z5362929

Tasks Undertaken/ Approaches

Author: Rohit Benjamin George

1. **Requirements Analysis:**

The project began with a thorough analysis of the systems requirements. This involved identifying the essential functionalities including managing patient demographics, diagnostic history, phenotypic traits, and genetic mutations. The requirements also emphasized robust search capabilities, secure role-based access, and the ability to generate individual and general reports exportable in PDF format. Additionally, the system was designed to ensure data privacy and regulatory compliance.

2. **Database Design:**

A normalised database schema was developed to store patient related data efficiently while ensuring integrity and scalability. The database was normalised to the Third Normal Form (3NF) to eliminate redundancy and ensure data consistency. The key table included: Patient, Phenotypes, Diagnostics, Mutations, Clinician, Reports, and Category. Each table was designed with appropriate primary and foreign keys to enforce relationships. There was constraints like NOT NULL, CHECK, and ENUM were applied to ensure data quality. For instance, the Sex column in the Patient table accepts only predefined values (Male, Female, Other), and the CategoryType column in the Category table is restricted to Diagnostics, Phenotypes, or Mutations.

3. **Web Functionality:**

The systems' functionality was implemented using HTML, PHP, JavaScript, AJAX and SQL. There was secure login and registration pages that were created to authenticate users and provide role-based access. A dynamic form was developed for adding and updating patient records including details like diagnostics, phenotypes and genetic mutations. Clinicians could also search for patients using criteria like name, diagnosis and categorize patients based on their medical conditions. Individual and general patient reports were dynamically generated, formatted and exported in PDF format.

4. **User Interface Design:**

The web interface had included CSS to provide a seamless experience for clinicians. There was a clear navigation menu that allowed users to move between functionalities like patient search, data entry and generating reports.

5. **Testing and Debugging:**

There was rigorous testing that was conducted to ensure the system's reliability such as each module such as patient insertion and PDF generation was tested independently to verify functionality. The interaction between the database, server-side scripts and frontend components was validated to ensure smooth data flow. There were common user errors like missing fields or incorrect input formats were identified and addressed through validation and error messages.

6. **Security and Data Integrity:**

There were security measures that were implemented to protect patient data such as users are authenticated based on roles e.g. clinician to ensure access to authorised functionalities. A UserActivity table was created to log login times, actions performed and logout times ensuring accountability.

7. **Documentation:**

The final phase involves preparing documentation and presenting the system through user manual where detailed instructions on how to use the system and sample login credentials are provided.

8. **Conclusion:**

The structured approach ensured the successful implementation of a system that demonstrated clinical workflow while maintaining data privacy. Through focusing on core functionalities, robust database design and user-friendly interfaces the project met its objectives effectively.

Bug Explanations and Attempts to Fix

Author: Vinoshan Parathan

1. **Secure Logout Feature**

Bug: There were times when the session key could be accessed by outside users for hijacking.

Fix: A logout script was implemented “logout.php” which when activated by the user, destroys the session and redirects the user back to the login page. Secure session cookies were put into place and regenerative session ID keys were used to secure the session management.

2. **Session Expiry and Redirection**

Bug: Unauthorised users were able to access pages without logging in.

Fix: Added session checks at the beginning of each file that redirects unauthorised users back to the login page.

3. **Login Form and Session Handling**

Bug: Users were able to view the database with unauthorised access due to session handling issues.

Fix: This was fixed by applying a session handling script at the beginning of each file such as “add_patient.php”, which ensured sessions were always status checked before allowing the user to access the sensitive data.

4. **SQL Injection Prevention**

Bug: Initially, basic SQL queries were being used which could have allowed for SQL injection vulnerabilities.

Fix: To fix this, raw SQL queries were converted to prepared statements by using “bind_param” to bind user inputs securely as seen in “add_patient.php”.

5. **Dynamic Navigation Links**

Bug: Initially, it was hard to handle navigation links with multiple patients, as the code would try to make a new PHP file for each clinician and patient. These new files were corrupt hence a new strategy needed to be introduced.

Fix: PHP variables were used to dynamically generate the URLs in navigation links, hence allowing pages to have the correct patient and clinician ID values passed.

6. **Generating PDFs with TCPDF**

Bug: Struggled to run basic code that would generate PDFs. The code used was elementary and did not use any plugins, addons, software to help generate the PDFs.

Fix: After Researching different programs and software to make the process simpler, TCPDF was found to be very useful and simple to understand. As seen in "generate_all_patients_pdf.php" the pdf creation was relatively simple after implementation of this open-source software.

7. **Missing Field Handing in Forms**

Bug: Optional Fields were not handled correctly, hence when a clinician would not have enough data, the patient could not be added.

Fix: Updated the form handling code to make sure only some fields that were essential were "required". NULL coalescing was also used for optional fields, ensuring the database could handle the fields correctly.

8. **Edit and Adding Multiple Entries**

Bug: When using only PHP to edit and add multiple phenotypes, mutations and diagnostics for patients, the data would not save into the proper SQL table. The code was also very messy.

Fix: By incorporating JavaScript to add fields while PHP saved the entries, the data would be saved correctly with less lines of code.

System User Manual

Author: Alice Gao

1. Creating an account and logging in

Upon the first access of any link of the website e.g. http://localhost/views/dashboard.php, the user will be taken to the Login page:

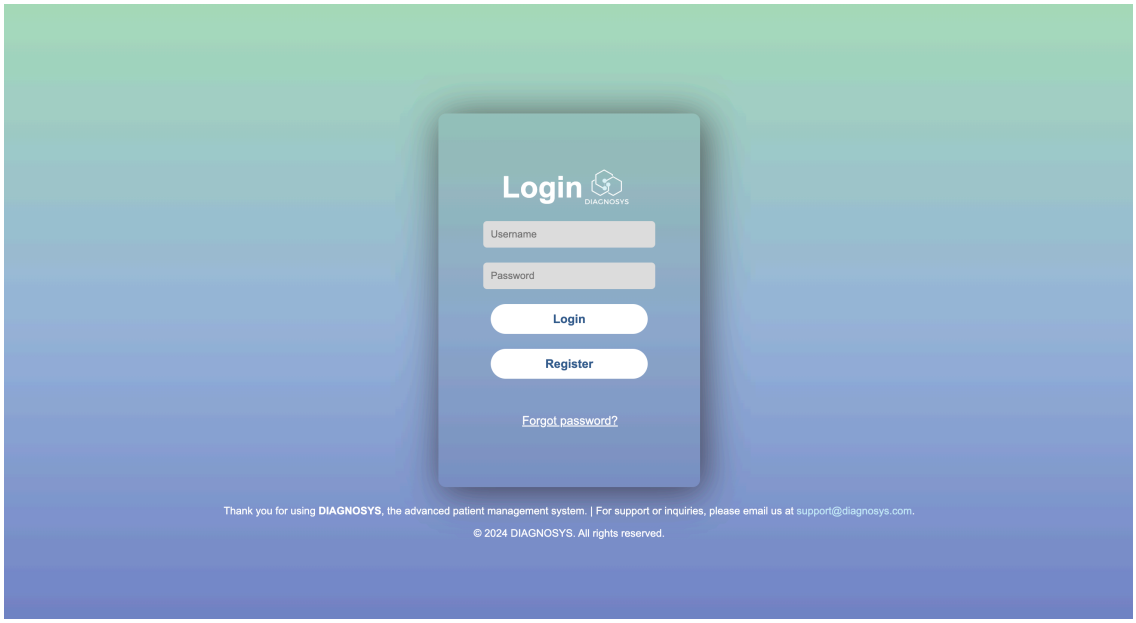

Figure 1: Login Page

The user will need to register for a Login as they do not have an account. Click the Register button.

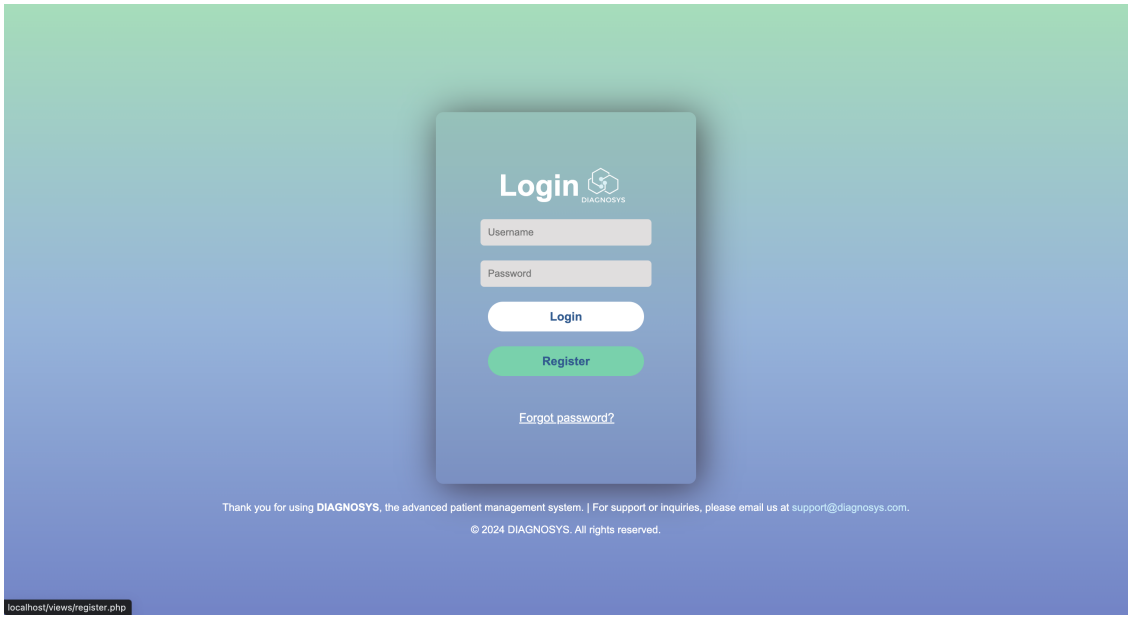

Figure 2: Hovering on the register button

The user will be taken to the registration page. If the user already has an account, they can click the ‘Already have an account?’ button to go back to the Login page.

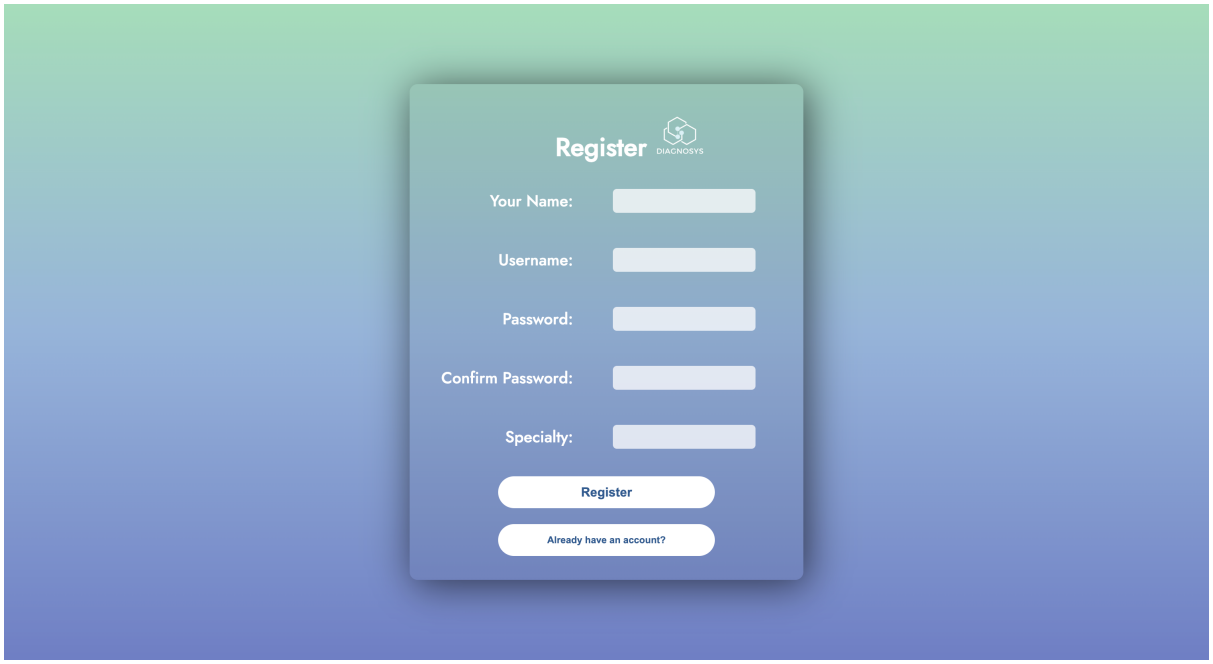

Figure 3: Registration page

The user will input their details, such as name and specialty, as well as creating a username and password. Inadequate fields (Figure 4) will show an error prompt and the user must follow its instructions and correct their entry to create a login. These fields maintain the validity of the input as well as to instil security measures.

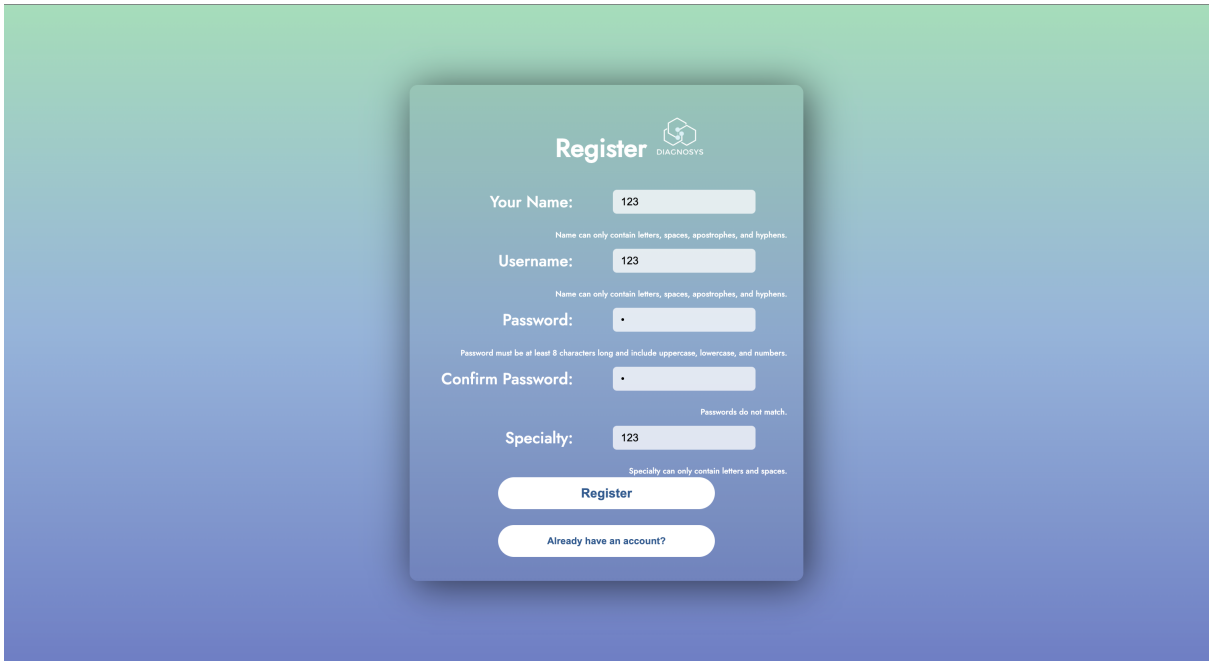

Figure 4: Demonstration of inadequate entered fields in the registration form. Prompts are shown for the user to action correct entries.

After entering valid inputs, the user will click on the Register button to create an account.

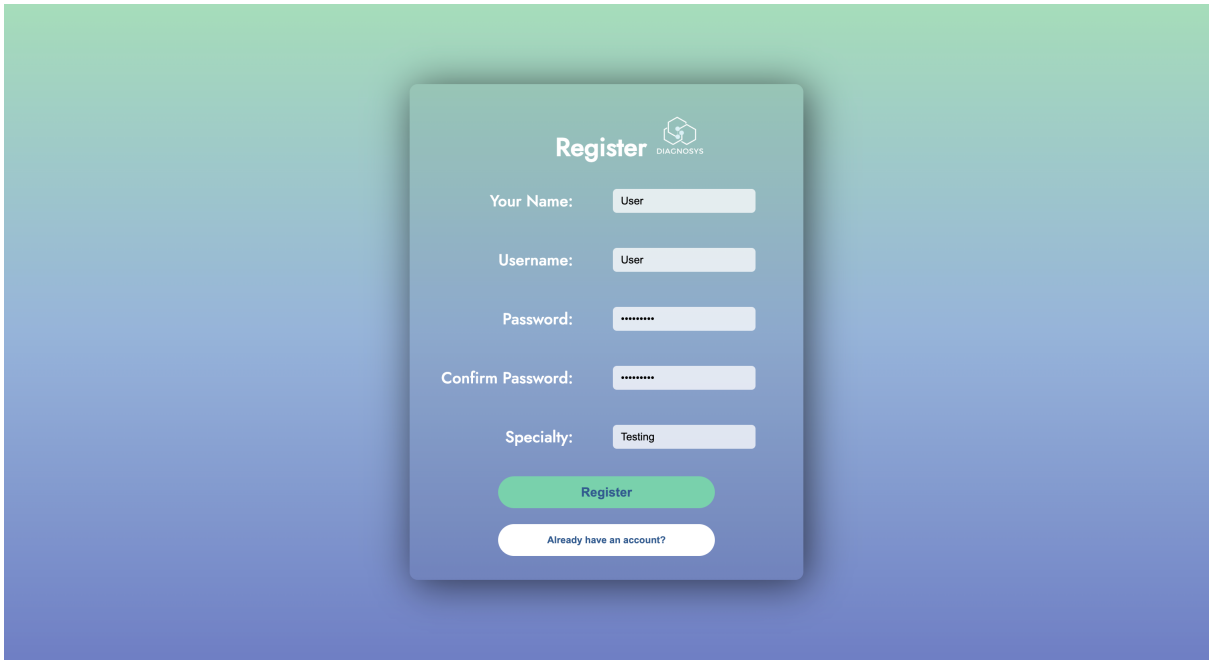

Figure 5: Hovering the Register button after valid field entries

After Registration, the user will be taken back to the Login page, where they will enter their account details and press Login to log in to the website.

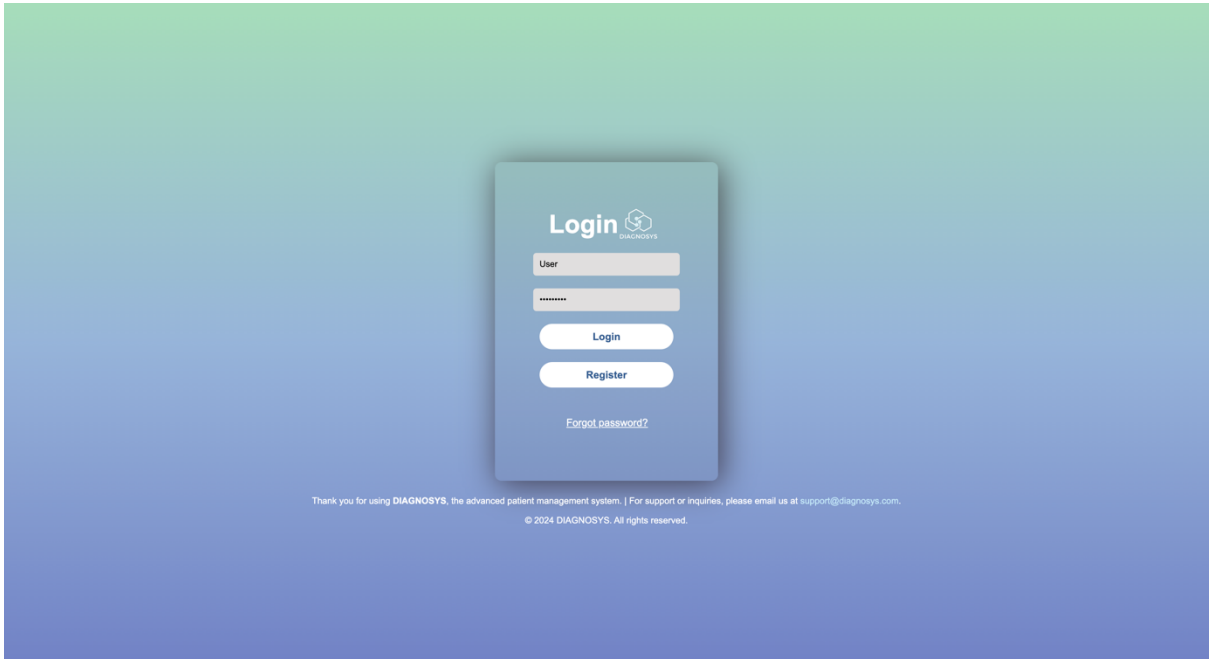

Figure 6: Entering login details

2. Navigating the Dashboard

Upon logging in, the user will be taken to the dashboard (figure 7), where they will be greeted and see available patient data presented in a table. They will also see a navigation bar with options to add new patient data or navigate to the About page to find out more about the system. The navigation bar also contains login information (username), as well as a logout button.

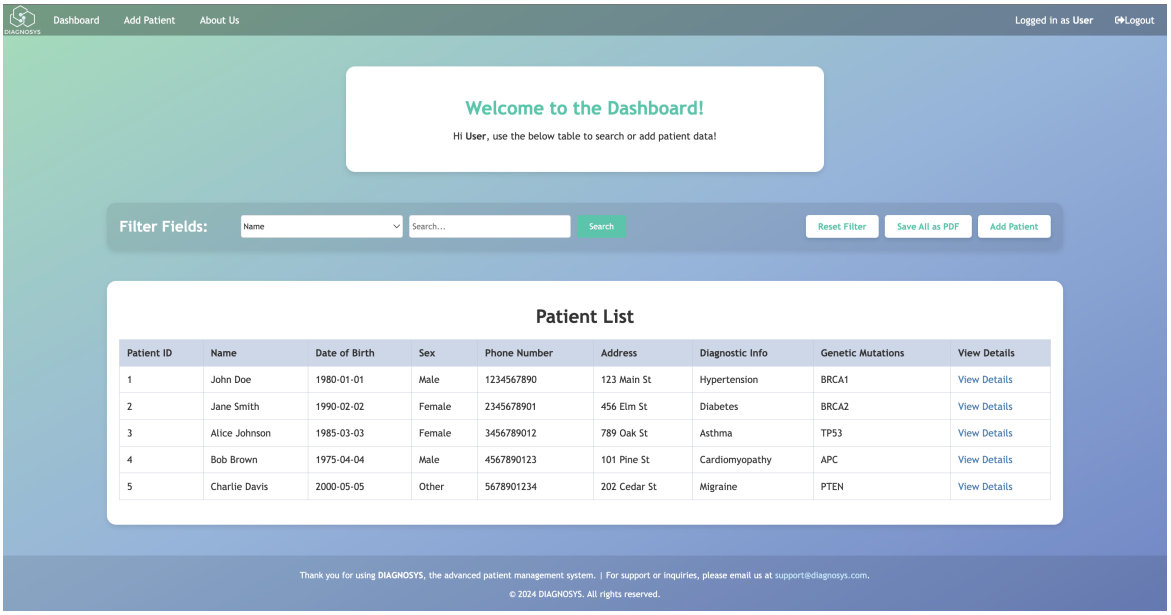

Figure 7: Dashboard

The user can filter patient data using the filter fields (Figure 8) above the patient data table, with fields like Name, DOB, sex, phone number, diagnostic info, and genetic mutations. Any fields containing the entered characters will be presented.

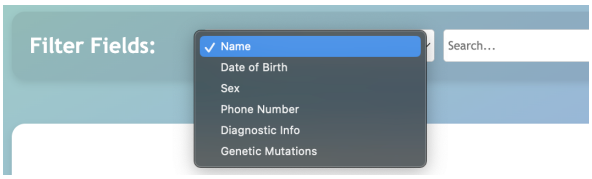

Figure 8: Filtering fields

An example of filtering by name is shown below:

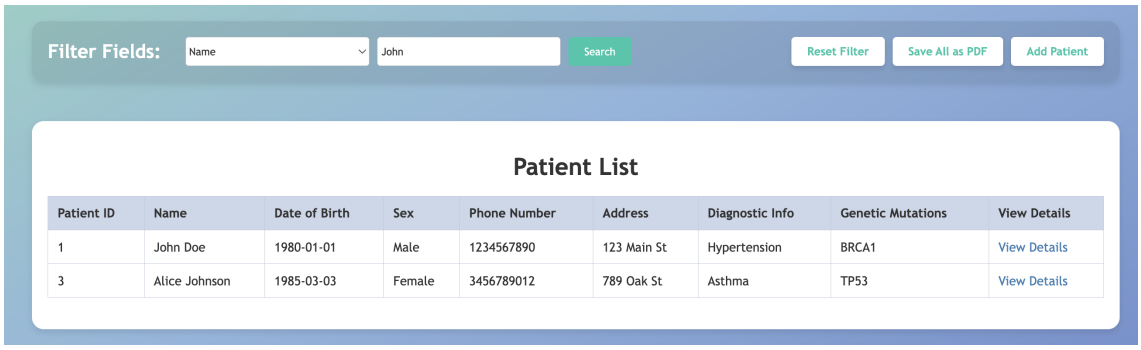

Figure 9: Filtering by name, example

An example of filtering by phone number is shown below:

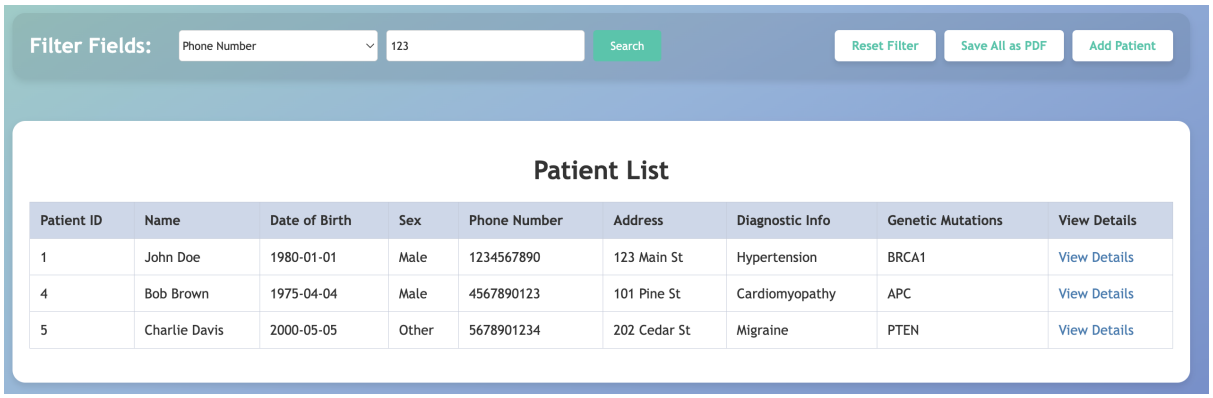

Figure 9: Filtering by phone number, example.

In addition, the user will be able to click Save All as PDF to save the list of filtered or current displayed patients in a PDF, as such:

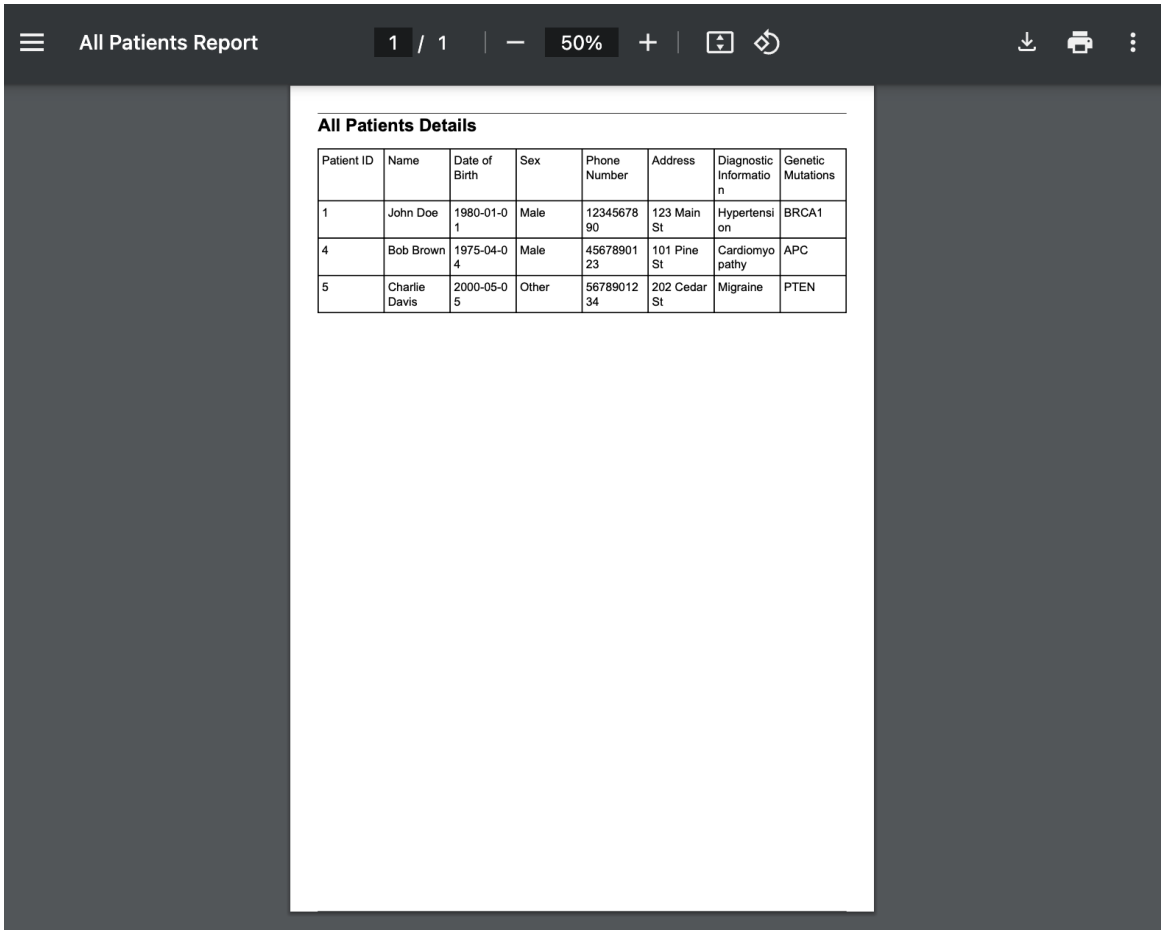

Figure 10: Filtered patient data saved in a PDF

3. Editing Patient Details

Next, the user will be able to click on View Details next to each patient, to view the patient's detailed profile.

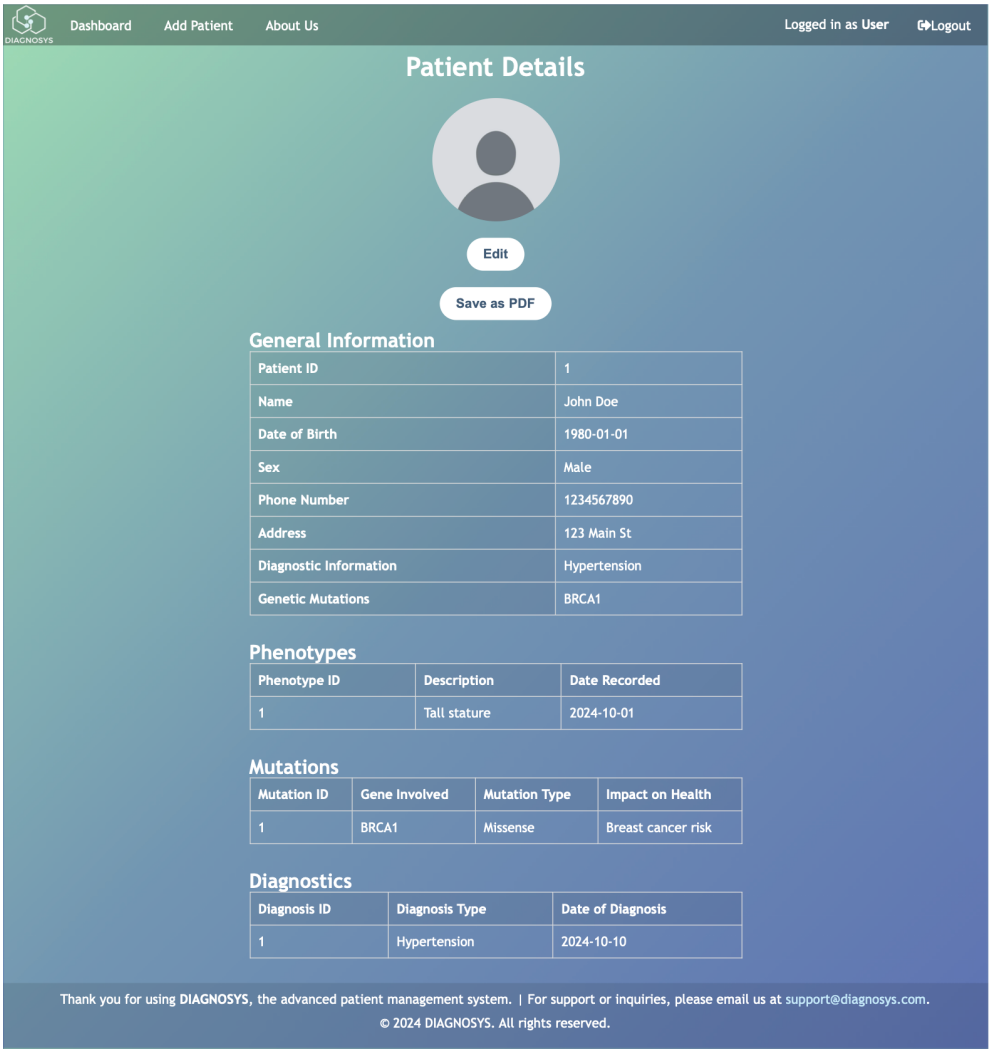

Figure 11: Example patient details

The user will be able to click on Edit, to edit the patient profile:

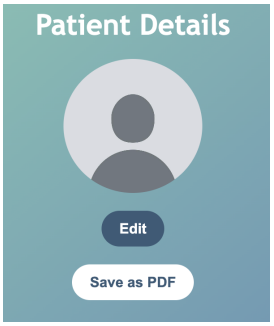

Figure 12: Hovering edit under patient details

Upon clicking edit, the user will be taken to an edit patient form to change or insert additional patient data.

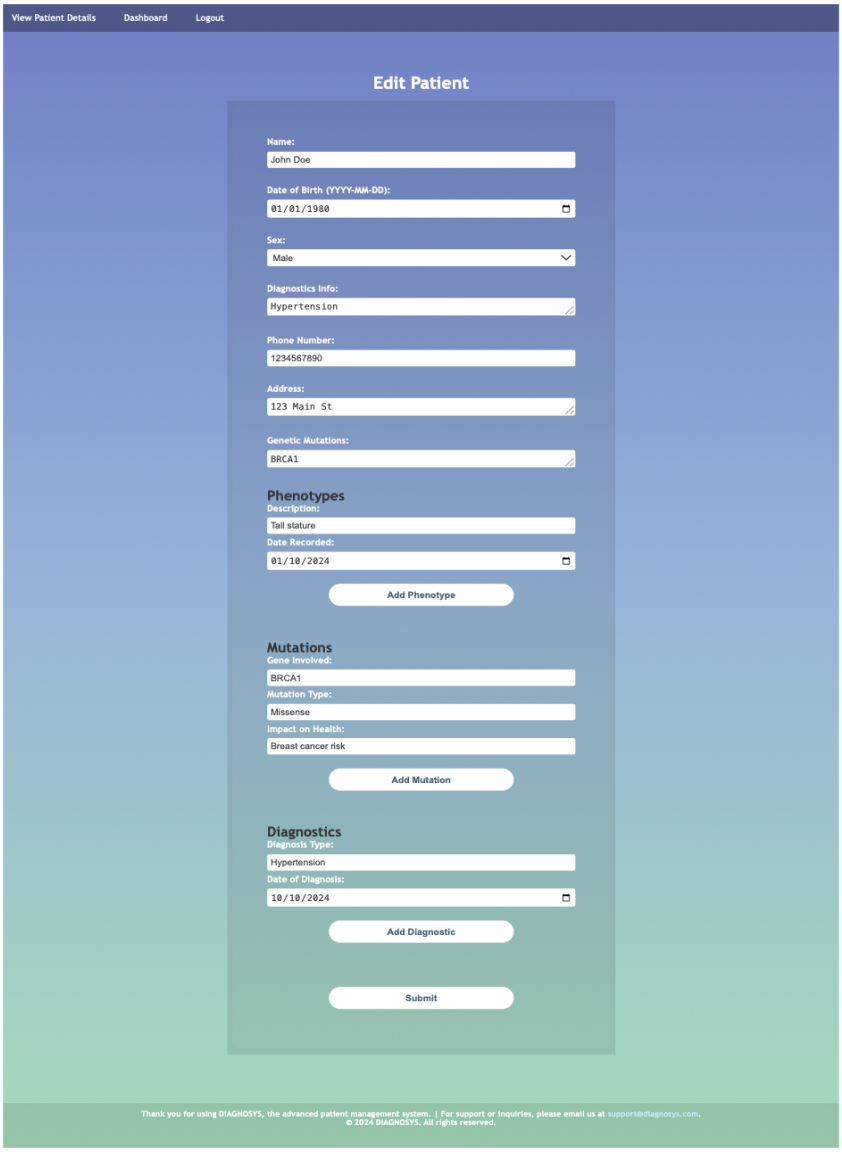

Figure 13: Edit patient form

Else, the user can click on save as PDF to save the patient details in a pdf:

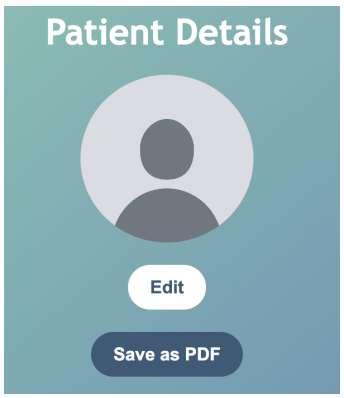

Figure 14: Hovering Save as PDF under patient details

As such:

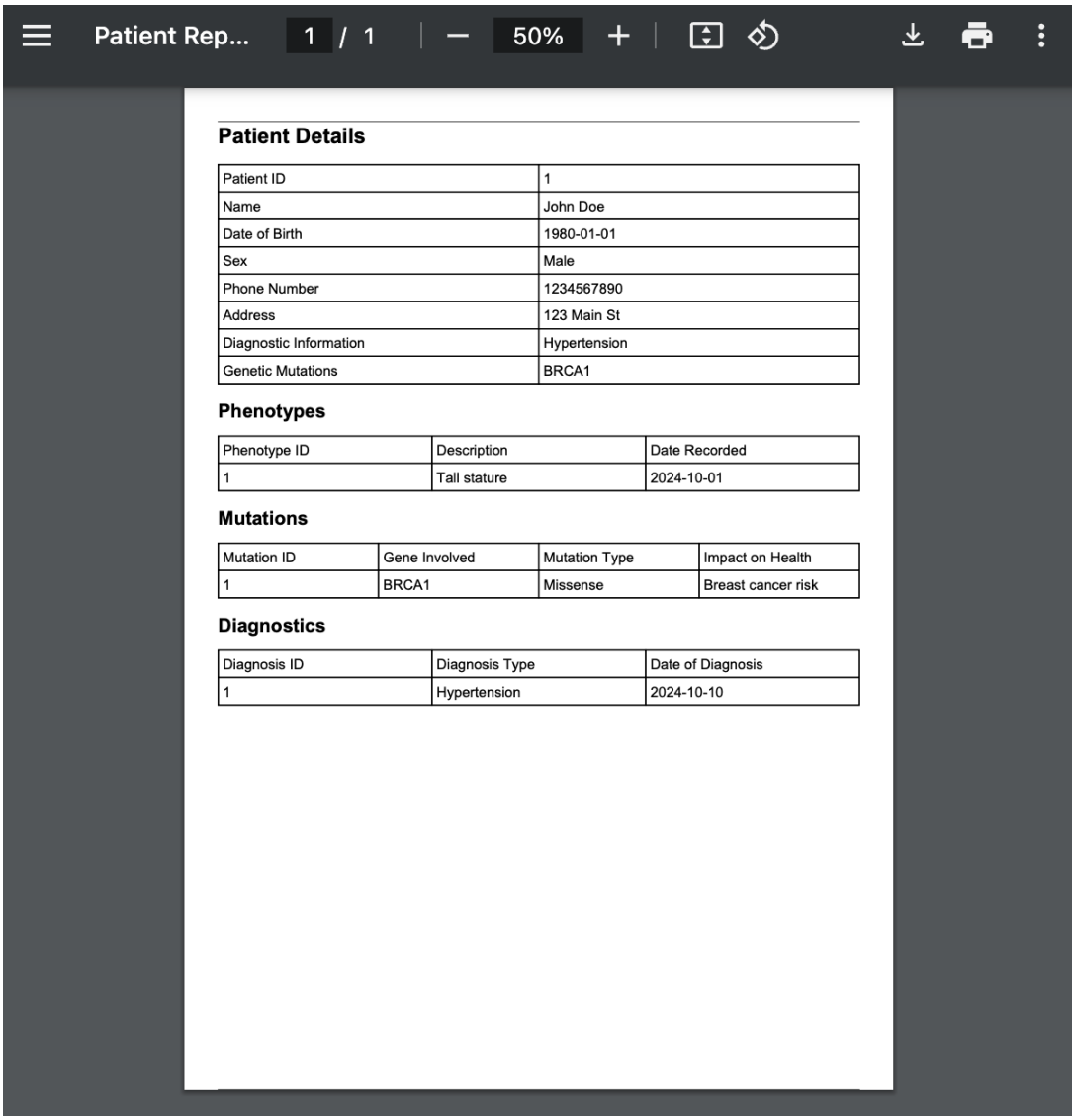

Figure 15: Downloading patient data as PDF

4. Adding Patient Data

Back to the dashboard, the user can either use the Add Patient button (Figure 16) above patient data to add a new patient or do so by clicking the button called Add Patient in the navigation bar.

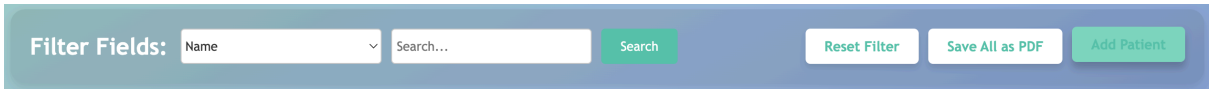

Figure 16: Hovering the Add Patient Button

Doing so will lead the user to the Add Patient form:

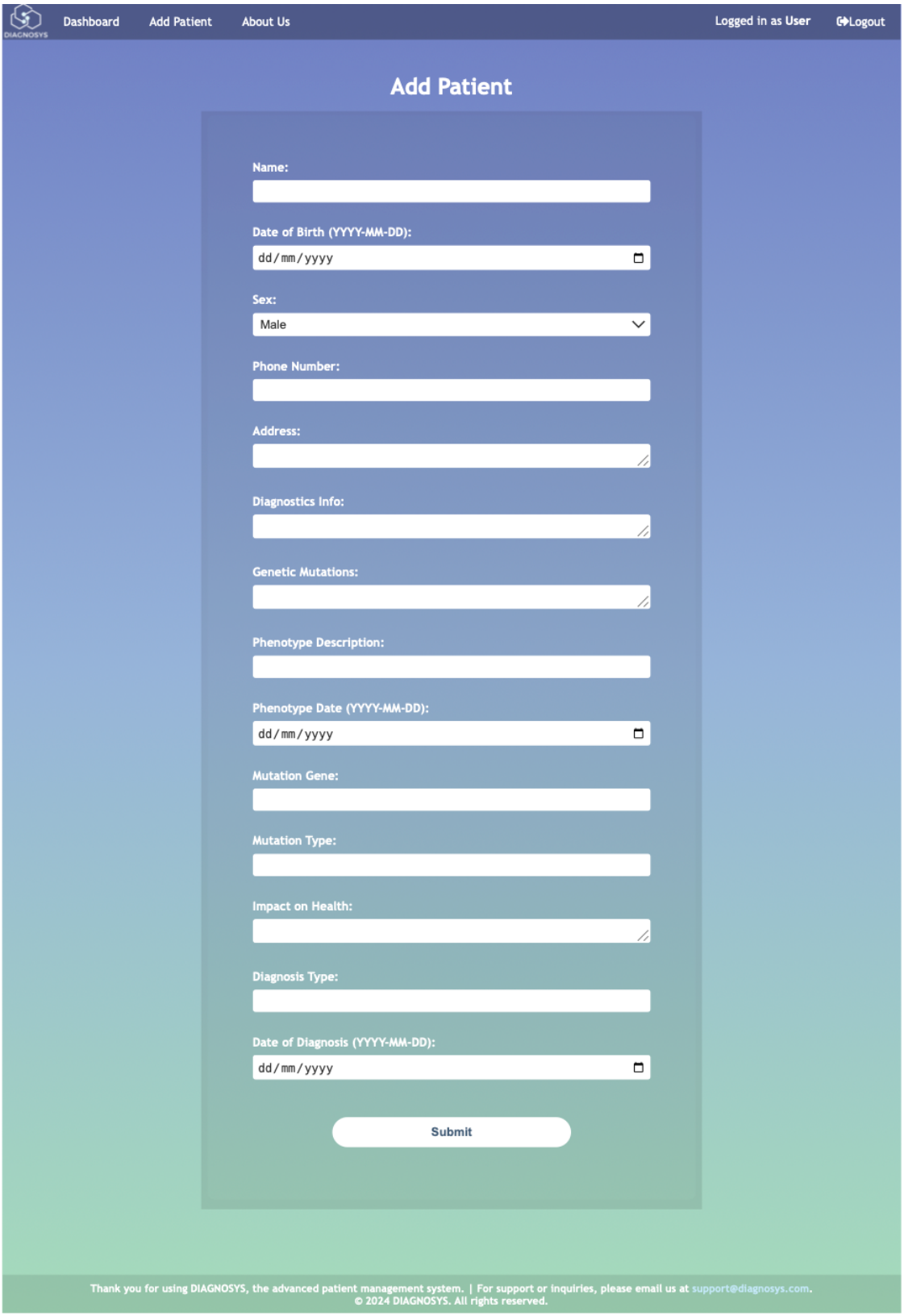

Figure 17: Add Patient Form, containing fields for patient diagnostic and genetic information

Fields like Patient name, DOB, Address, Diagnostic Info, Genetic Mutations and Phenotype Description, are required fields and must have an input. If the field is left blank, a pop up will notify the user to include this information.

Figure 18: Required field left blank, pop up message

Moreover, if the date field is incomplete, another pop up will notify the user to amend.

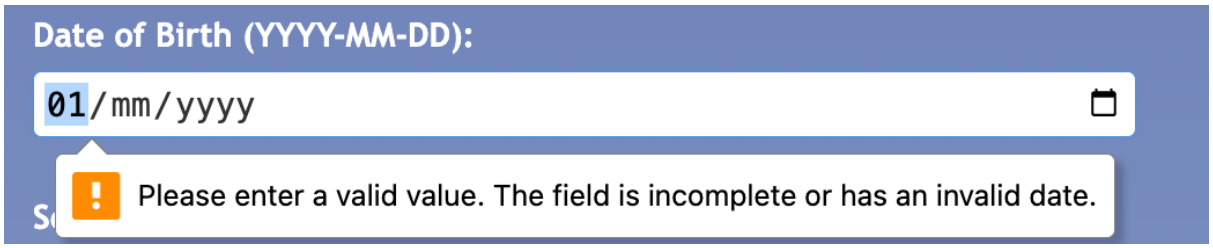

Figure 19: Date of birth incomplete, pop up message

5. About Us and Logging Out

In the navigation bar, the user can click on About Us to learn more about the system or click on Logout if they would like to exit the session.

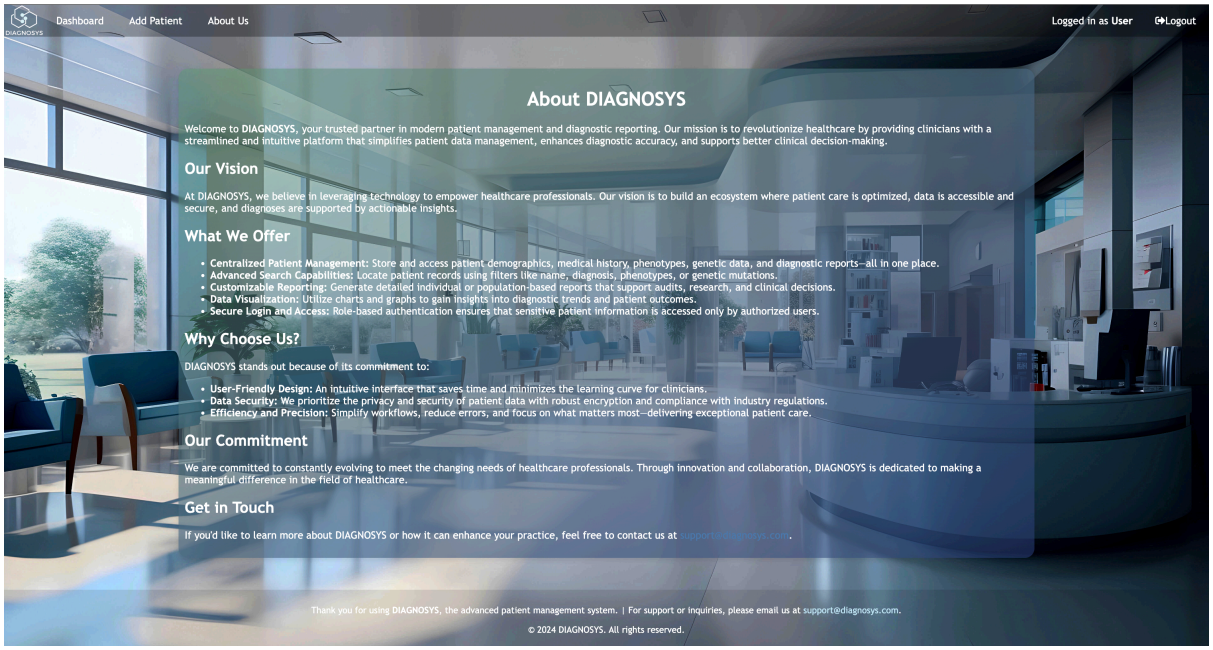

Figure 20: The About Us page, providing detail on the system

Figure 21: Logout button to exit and sign out of the session