SCU

SLIIT CITY UNI

ETF - ASSIGNMENT 02

Patient Management System

Group: 13
Date: 12/06/2025

Group details

SA No.	Name of the Member	Work Done by Member
SA24610652	**M. A. Pathirana**	Back End (REST API) with Node
SA24610735	**J. A. N. C. Jayakody**	Ajax (Fetch data from the server)
SA24610653	**H. A. S. Shevinu**	Front End + jQuery
SA24610678	**A. W. M. T Henuka**	Back End (REST API) with Node
SA24610733	**G. T. K. Silva**	Front End + jQuery
SA24610751	**B. B. D. Kumara**	Ajax (Fetch data from the server)

Content Page

1. Project Overview

Purpose of the system.

This project is a web based Patient Management System for a hospital. Developed for simplifying and digitalizing the process of handling patient records more effectively and efficiently. Allows the users of the system to add, search, view and update/delete patient records through an interactive user friendly UI.

Technologies implemented for this project:

1. Front-End
   - **HTML5** - Structure of all web pages
   - **CSS3 / Bootstrap 5.3** - Styling and responsive layout
   - **jQuery 3.7** - DOM manipulation, event handling, data validation.
   - **AJAX** - Handles asynchronous data fetching from the server.
   - **NodeJS** - Manages the REST API and client-server communication.

2. Data Handling
   - **JSON -** Used for exchanging patient data with the REST API
   - **REST API -** Allows to perform operations like retrieving, updating, or deleting patient data.
   - **MongoDB -** NoSQL database used to store the patient data.

3. User Interface Enhancements
   - **Bootstrap Components** – Navbar, buttons, tables, forms
   - **Dark Mode Toggle** – Implemented using jQuery
   - **Form Validation** – Real-time input validation with jQuery

2. Guide to add a new service

***Note**: Steps on how to implement a new feature to search by 'Status' using the existing code.*

Backend Implementation

Added support for querying patients by Status along with other parameters (PID, FirstName, LastName, Email, NearCity, Doctor, Guardian). The /searchpatients endpoint dynamically builds the query and retrieves matching records from the database.

```
app.get("/searchpatients", async (req, res) => {
  const query = {};
  if (req.query.PID) query.PID = req.query.PID;
  if (req.query.FirstName) query.FirstName = req.query.FirstName;
  if (req.query.LastName) query.LastName = req.query.LastName;
  if (req.query.Email) query.Email = req.query.Email;
  if (req.query.NearCity) query.NearCity = req.query.NearCity;
  if (req.query.Doctor) query.Doctor = req.query.Doctor;
  if (req.query.Guardian) query.Guardian = req.query.Guardian;
  if (req.query.Status) query.Status = req.query.Status;
  try {
    const patients = await PatientModel.find(query).lean();
    res.json(patients);
  } catch (err) {
    res.status(500).send('Error searching patients: ' + err.message);
  }
});
```

Frontend

A new "By Status" option has been added to the searchType dropdown, which allows users to filter patients by status.

```
<div class="col-md-4">
<label for="searchType" class="form-label">Search by</label>
<select class="form-select" id="searchType" name="searchType">
      <option selected disabled>Select option</option>
      <option value="pid">By Patient ID</option>
      <option value="firstName">By First Name</option>
      <option value="lastName">By Last Name</option>
      <option value="email">By Email</option>
      <option value="doctor">By Doctor</option>
      <option value="NearestCity">By Nearest City</option>
      <option value="guardian">Guardian</option>
      <option value="status">By Status</option>
</select>
</div>
```

Ajax

The JavaScript switch-case now includes a status case to handle the new search parameter

```
let queryParam = '';
switch (searchType) {
      case 'pid': queryParam = 'PID=' + encodeURIComponent(searchValue);
      Break;
      case 'firstName': queryParam = 'FirstName=' +
encodeURIComponent(searchValue);
      break;
      case 'lastName': queryParam = 'LastName=' +
encodeURIComponent(searchValue);
      break;
      case 'email': queryParam = 'Email=' +
encodeURIComponent(searchValue);
      break;
      case 'doctor': queryParam = 'Doctor=' +
encodeURIComponent(searchValue);
      break;
      case 'NearestCity': queryParam = 'NearCity=' +
encodeURIComponent(searchValue);
      break;
      case 'guardian': queryParam = 'Guardian=' +
encodeURIComponent(searchValue);
      Break;
      case 'status': queryParam = 'Status=' +
encodeURIComponent(searchValue);
      Break;
      }
```