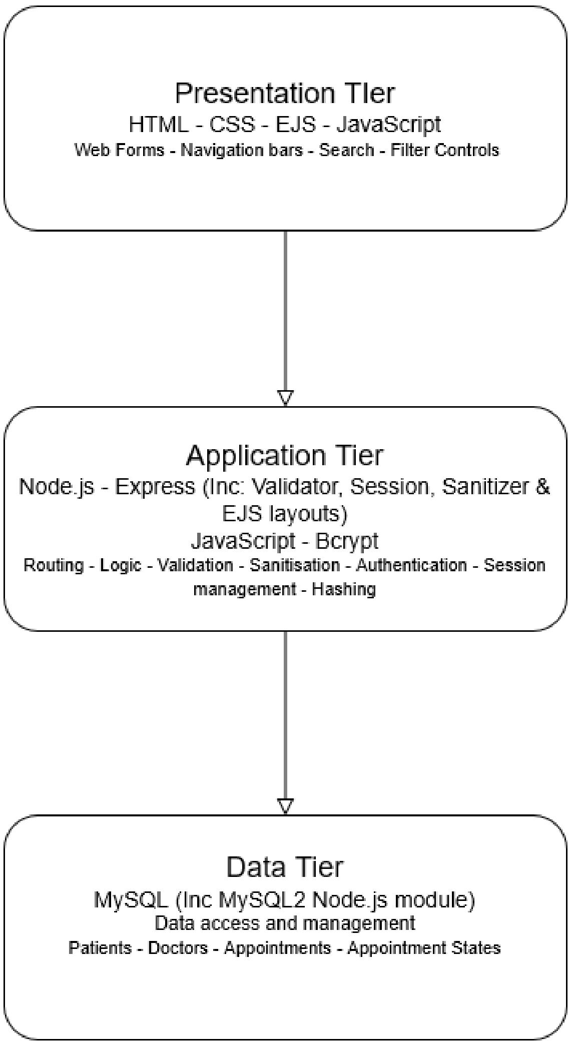
Report: GP Booking Management Application

Outline

GP (General Practice) Appointment Manager is an application for scheduling appointments. Designed for small GPs, especially ones in rural area where employee numbers and resources are limited. In those situations, clinical staff often have to take on the role of administrative staff too. This system allows users (patients) to register for an account, request an appointment, view their appointment history and cancel upcoming appointments. Administrative users can view and edit appointment records including changing the appointment date, assign doctors to appointments and change the appointment status. They can also view and search the database of patient records.

Architecture

This project has a 3-tier architecture.


Presentation TIer
HTML - CSS - EJS - JavaScript
Web Forms - Navigation bars - Search - Filter Controls
Application Tier
Node.js - Express (Inc: Validator, Session, Sanitizer & EJS layouts)
JavaScript - Bcrypt
Routing - Logic - Validation - Sanitisation - Authentication - Session management - Hashing
Data Tier
MySQL (Inc MySQL2 Node.js module)
Data access and management
Patients - Doctors - Appointments - Appointment States

Application Tier

- Node.js and Express control HTTP requests through route handlers, with each one controlling a separate routing group.
- Controls logic and calls between data and presentation tier.
- Input validation and sanitisation using Express validator.
- Bcrypt for authentication and hashing.

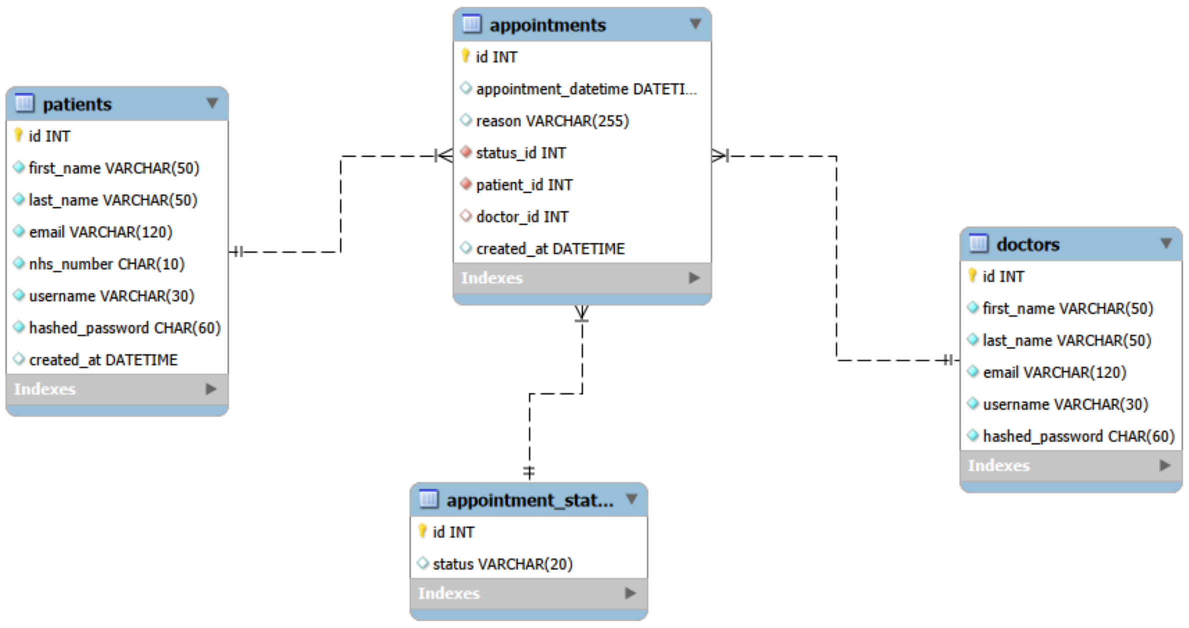
Data Tier

- MySQL for managing the database.
- The client is accessed using MySQL2 module for Node.js, allowing queries to be actioned through JavaScript code.
- Data access is encapsulated in the model modules.

Presentation Tier

- EJS for rendered HTML.
- Layouts and render partials using 'express-ejs-layouts' enabling navbar rending and template pages.
- CSS for styling.

Data Model



Model uses a relational database schema, with 3 base tables and one lookup table.

Patients store personal information and authentication details of the health service users. The doctors table stores similar information for staff.

Appointments stores factual information about an appointment such as the date and time of the appointment and reason. It requires a patient ID and a doctor ID both foreign keys to their relevant tables. Status ID references the ID of a matching status on appointment states.

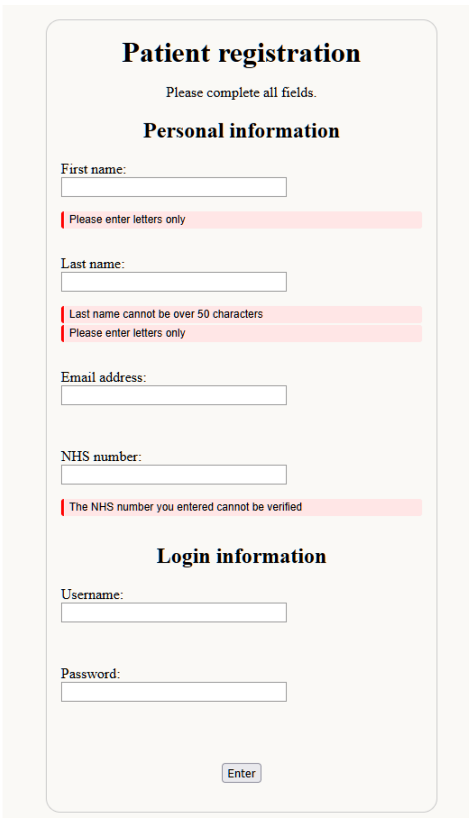
Appointment states is a lookup table storing a static list of appointment statuses, to be referenced by the appointments table.

User Functionality

The application includes two user views, one for health service users (patients) and one for staff (doctors).

Patients Functionality

Registration



Patient registration

Please complete all fields.

Personal information

First name:

Please enter letters only

Last name:

Last name cannot be over 50 characters

Please enter letters only

Email address:

NHS number:

The NHS number you entered cannot be verified

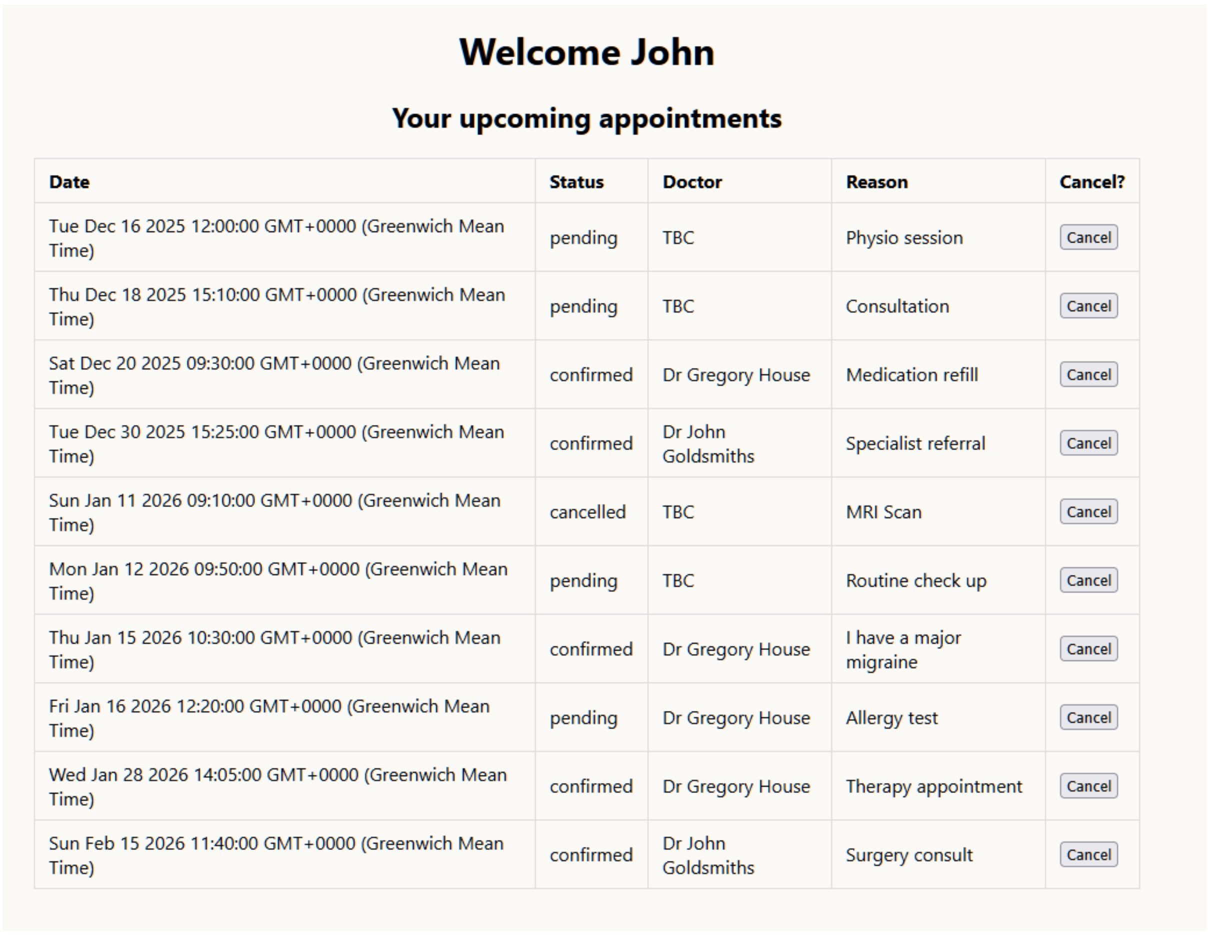
Login information

Username:

Password:

To access the appointment bookings, patients can register for an account on the registration page. Helpful error messages are displayed if their credentials are invalid.

Patients Portal



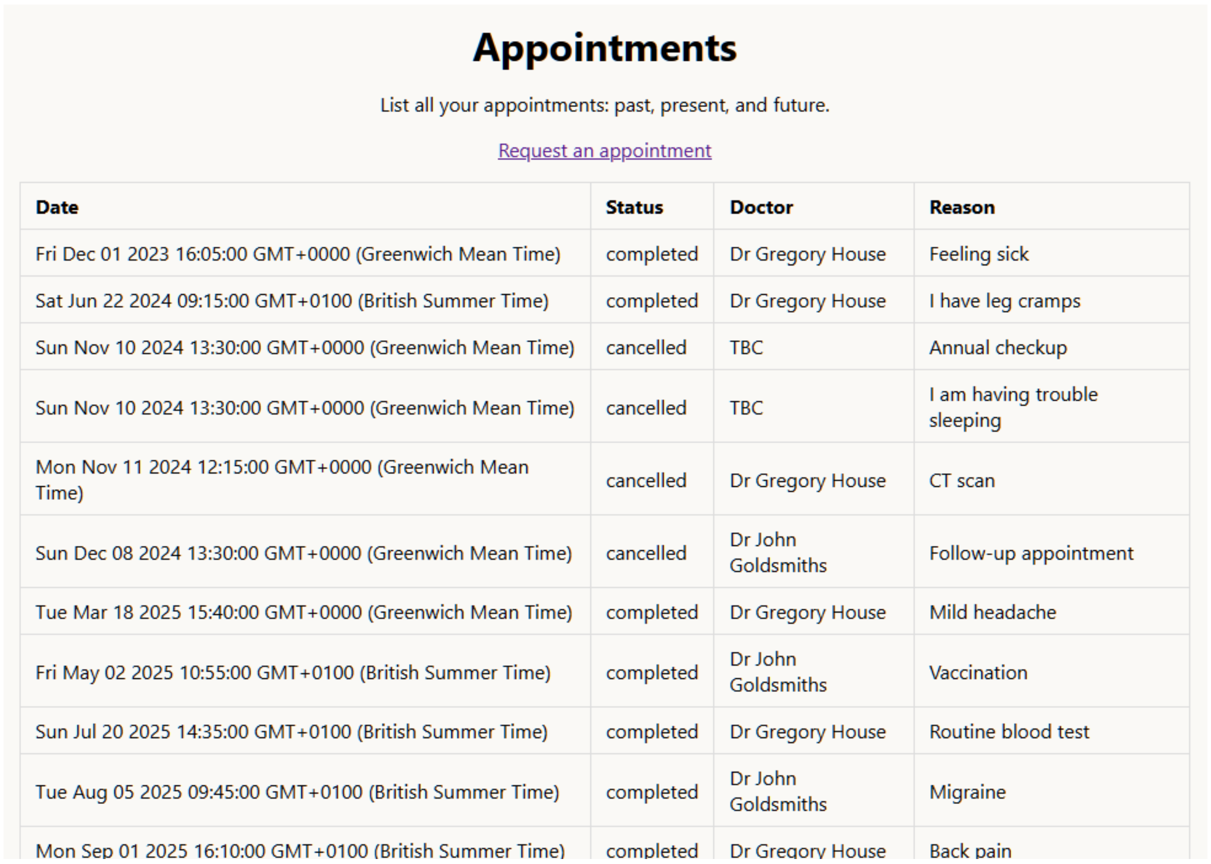
Welcome John

Your upcoming appointments

Date	Status	Doctor	Reason	Cancel?
Tue Dec 16 2025 12:00:00 GMT+0000 (Greenwich Mean Time)	pending	TBC	Physio session	Cancel
Thu Dec 18 2025 15:10:00 GMT+0000 (Greenwich Mean Time)	pending	TBC	Consultation	Cancel
Sat Dec 20 2025 09:30:00 GMT+0000 (Greenwich Mean Time)	confirmed	Dr Gregory House	Medication refill	Cancel
Tue Dec 30 2025 15:25:00 GMT+0000 (Greenwich Mean Time)	confirmed	Dr John Goldsmiths	Specialist referral	Cancel
Sun Jan 11 2026 09:10:00 GMT+0000 (Greenwich Mean Time)	cancelled	TBC	MRI Scan	Cancel
Mon Jan 12 2026 09:50:00 GMT+0000 (Greenwich Mean Time)	pending	TBC	Routine check up	Cancel
Thu Jan 15 2026 10:30:00 GMT+0000 (Greenwich Mean Time)	confirmed	Dr Gregory House	I have a major migraine	Cancel
Fri Jan 16 2026 12:20:00 GMT+0000 (Greenwich Mean Time)	pending	Dr Gregory House	Allergy test	Cancel
Wed Jan 28 2026 14:05:00 GMT+0000 (Greenwich Mean Time)	confirmed	Dr Gregory House	Therapy appointment	Cancel
Sun Feb 15 2026 11:40:00 GMT+0000 (Greenwich Mean Time)	confirmed	Dr John Goldsmiths	Surgery consult	Cancel

Once logged in, patients can view or cancel their upcoming appointments.

Appointment List



Appointments

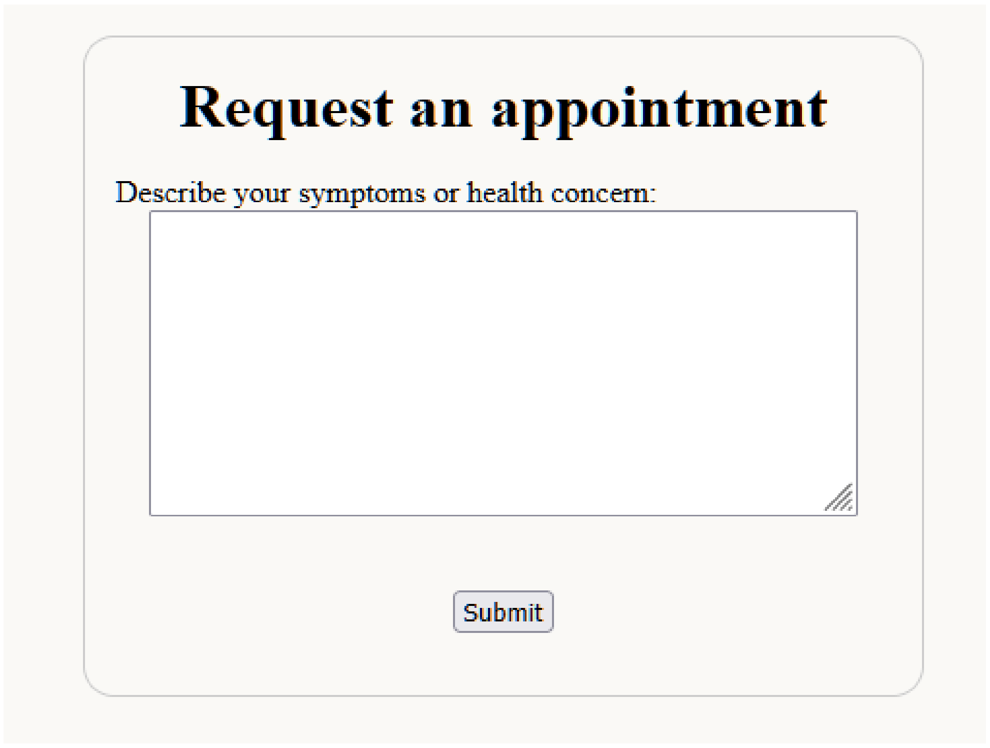
List all your appointments: past, present, and future.

Request an appointment

Date	Status	Doctor	Reason
Fri Dec 01 2023 16:05:00 GMT+0000 (Greenwich Mean Time)	completed	Dr Gregory House	Feeling sick
Sat Jun 22 2024 09:15:00 GMT+0100 (British Summer Time)	completed	Dr Gregory House	I have leg cramps
Sun Nov 10 2024 13:30:00 GMT+0000 (Greenwich Mean Time)	cancelled	TBC	Annual checkup
Sun Nov 10 2024 13:30:00 GMT+0000 (Greenwich Mean Time)	cancelled	TBC	I am having trouble sleeping
Mon Nov 11 2024 12:15:00 GMT+0000 (Greenwich Mean Time)	cancelled	Dr Gregory House	CT scan
Sun Dec 08 2024 13:30:00 GMT+0000 (Greenwich Mean Time)	cancelled	Dr John Goldsmiths	Follow-up appointment
Tue Mar 18 2025 15:40:00 GMT+0000 (Greenwich Mean Time)	completed	Dr Gregory House	Mild headache
Fri May 02 2025 10:55:00 GMT+0100 (British Summer Time)	completed	Dr John Goldsmiths	Vaccination
Sun Jul 20 2025 14:35:00 GMT+0100 (British Summer Time)	completed	Dr Gregory House	Routine blood test
Tue Aug 05 2025 09:45:00 GMT+0100 (British Summer Time)	completed	Dr John Goldsmiths	Migraine
Mon Sep 01 2025 16:10:00 GMT+0100 (British Summer Time)	completed	Dr Gregory House	Back pain

Page displays all appointments connected to the health service user.

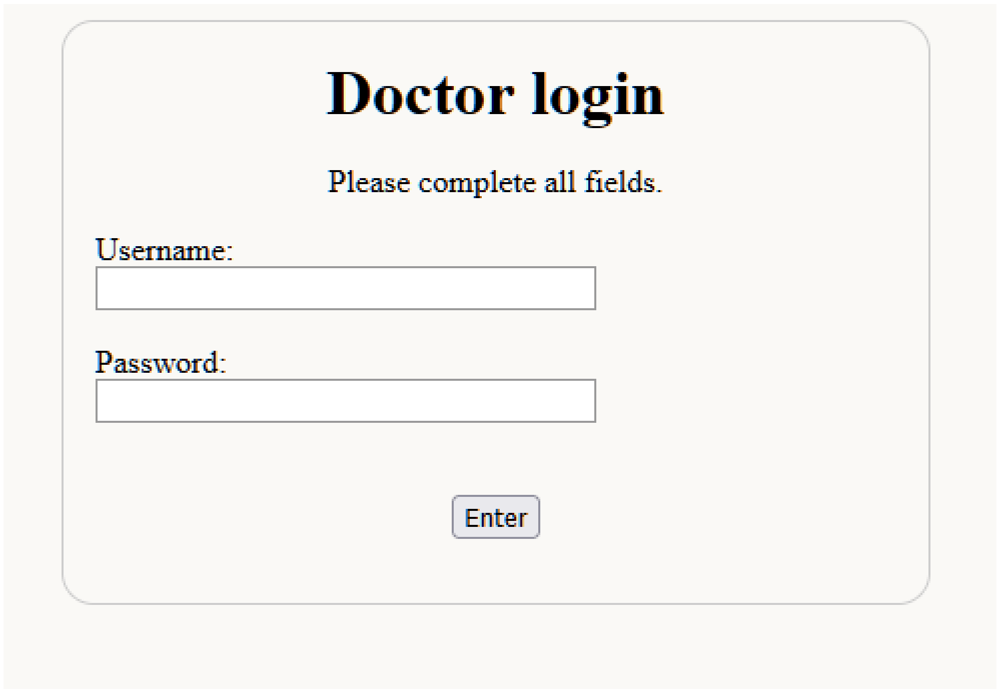
Appointment Request



This allows the user to send a request, simply by entering a reason for their request. Upon submission it creates an appointment in an unassigned state (status set to pending, date and doctor left blank. Its simple design is based on personal experience with GPs, they tend to only allow users to request an appointment but not specify a date, time or specific doctor. Then once a request is made they tend to receive an appointment date and time details without any input from themselves.

Staff Functionality

Login



Doctor login

Please complete all fields.

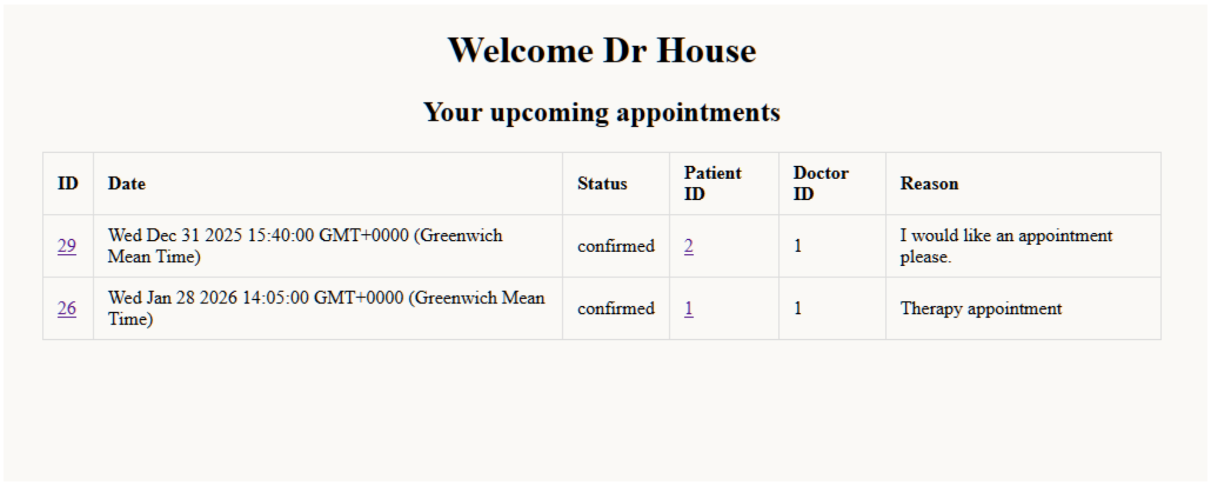
Username:

Password:

Enter

Doctors do not need to register to the application, instead, they are added to the database directly. Their login page does not have a navbar so that administrative tools are not exposed to the public.

Dashboard



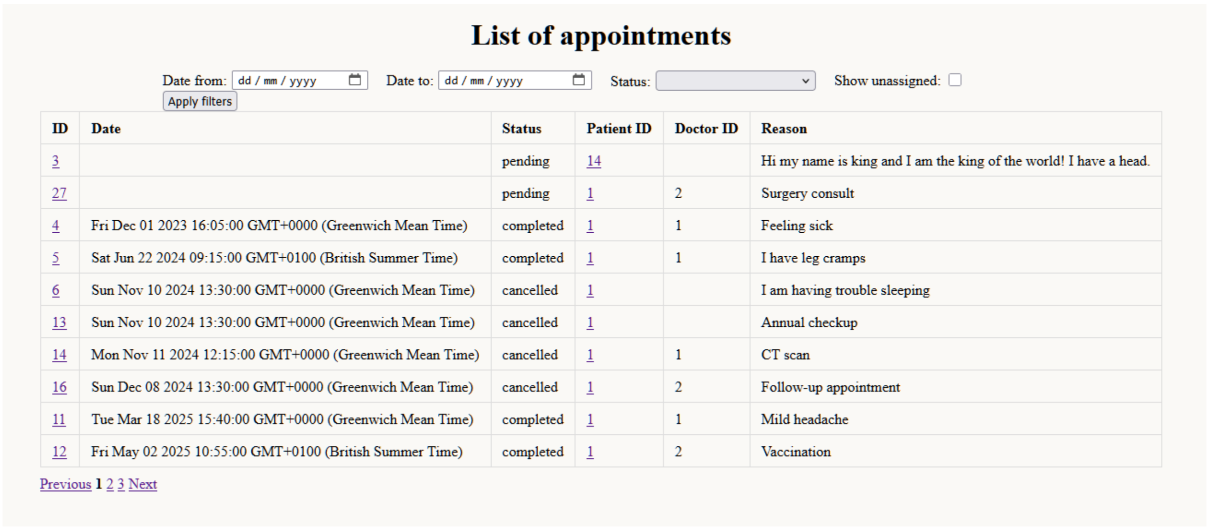
Welcome Dr House

Your upcoming appointments

ID	Date	Status	Patient ID	Doctor ID	Reason
29	Wed Dec 31 2025 15:40:00 GMT+0000 (Greenwich Mean Time)	confirmed	2	1	I would like an appointment please.
26	Wed Jan 28 2026 14:05:00 GMT+0000 (Greenwich Mean Time)	confirmed	1	1	Therapy appointment

Displays upcoming appointments that are assigned to the doctor that logged in.

Appointments Table



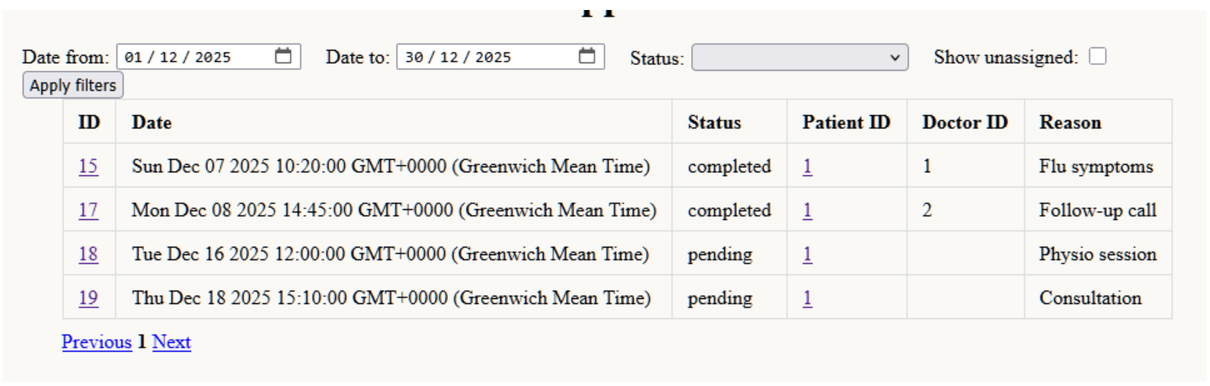
List of appointments

Date from: dd / mm / yyyy Date to: dd / mm / yyyy Status: Show unassigned: ☐
Apply filters

ID	Date	Status	Patient ID	Doctor ID	Reason
3		pending	14		Hi my name is king and I am the king of the world! I have a head.
27		pending	1	2	Surgery consult
4	Fri Dec 01 2023 16:05:00 GMT+0000 (Greenwich Mean Time)	completed	1	1	Feeling sick
5	Sat Jun 22 2024 09:15:00 GMT+0100 (British Summer Time)	completed	1	1	I have leg cramps
6	Sun Nov 10 2024 13:30:00 GMT+0000 (Greenwich Mean Time)	cancelled	1		I am having trouble sleeping
13	Sun Nov 10 2024 13:30:00 GMT+0000 (Greenwich Mean Time)	cancelled	1		Annual checkup
14	Mon Nov 11 2024 12:15:00 GMT+0000 (Greenwich Mean Time)	cancelled	1	1	CT scan
16	Sun Dec 08 2024 13:30:00 GMT+0000 (Greenwich Mean Time)	cancelled	1	2	Follow-up appointment
11	Tue Mar 18 2025 15:40:00 GMT+0000 (Greenwich Mean Time)	completed	1	1	Mild headache
12	Fri May 02 2025 10:55:00 GMT+0100 (British Summer Time)	completed	1	2	Vaccination

Previous 1 2 3 Next

Displays all appointments in the database. Unallocated appointments (pending and without a date) are displayed at the top as these are new requests that need to be processed, then it is ordered by date ascending and rows without a date but are not pending are last (usually cancelled appointments). Results are paginated and there are several filters that can be applied.

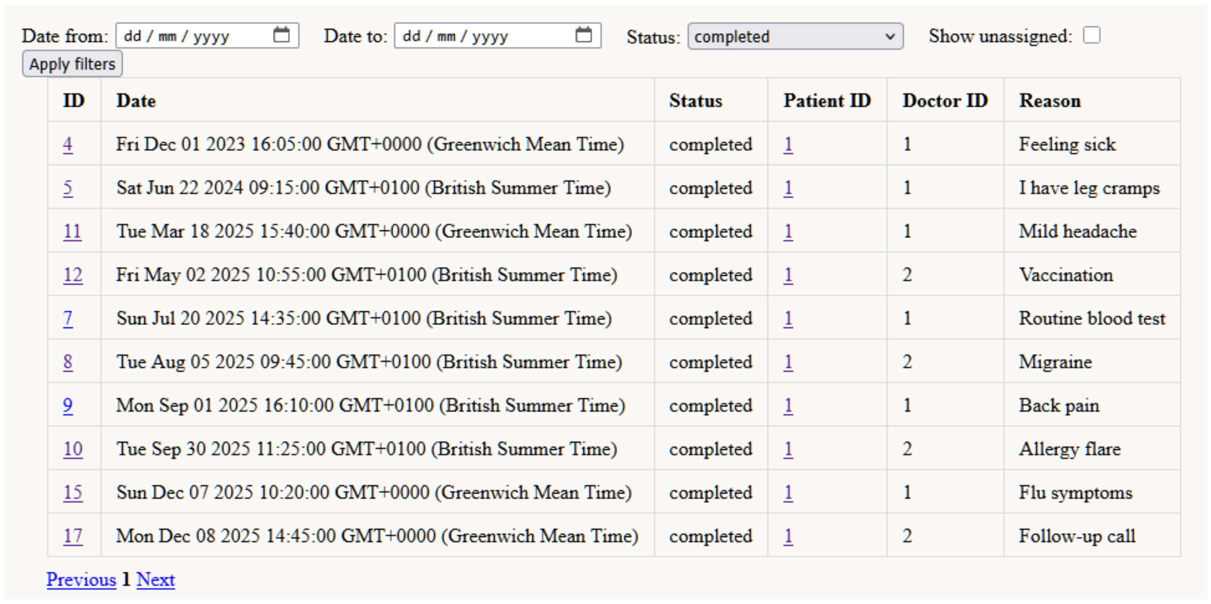


Date from: 01 / 12 / 2025 Date to: 30 / 12 / 2025 Status: Show unassigned: ☐
Apply filters

ID	Date	Status	Patient ID	Doctor ID	Reason
15	Sun Dec 07 2025 10:20:00 GMT+0000 (Greenwich Mean Time)	completed	1	1	Flu symptoms
17	Mon Dec 08 2025 14:45:00 GMT+0000 (Greenwich Mean Time)	completed	1	2	Follow-up call
18	Tue Dec 16 2025 12:00:00 GMT+0000 (Greenwich Mean Time)	pending	1		Physio session
19	Thu Dec 18 2025 15:10:00 GMT+0000 (Greenwich Mean Time)	pending	1		Consultation

Previous 1 Next

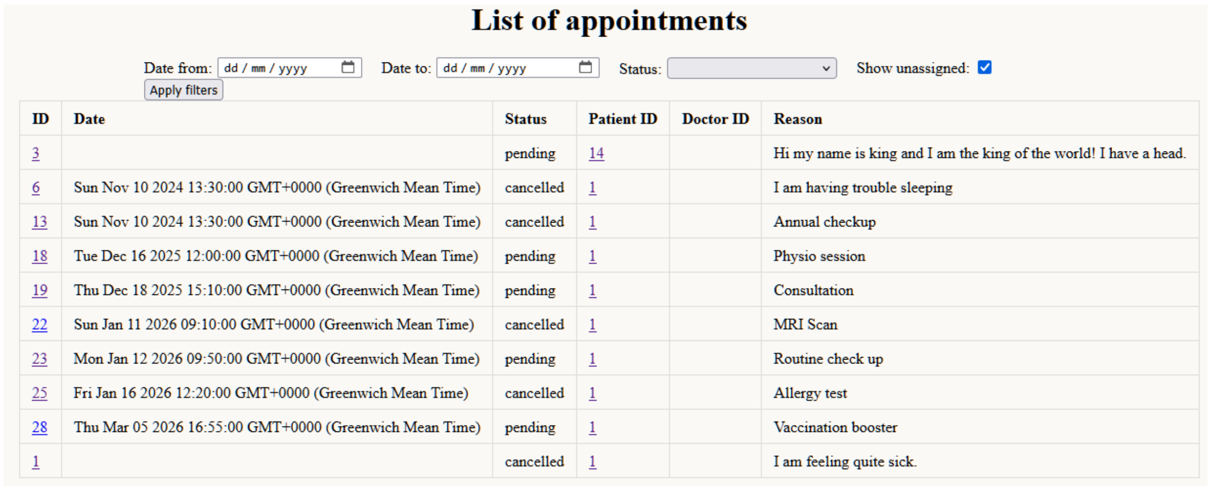
Date filter.


Date from: dd / mm / yyyy   Date to: dd / mm / yyyy   Status: completed   Show unassigned: ☐
Apply filters

ID	Date	Status	Patient ID	Doctor ID	Reason
4	Fri Dec 01 2023 16:05:00 GMT+0000 (Greenwich Mean Time)	completed	1	1	Feeling sick
5	Sat Jun 22 2024 09:15:00 GMT+0100 (British Summer Time)	completed	1	1	I have leg cramps
11	Tue Mar 18 2025 15:40:00 GMT+0000 (Greenwich Mean Time)	completed	1	1	Mild headache
12	Fri May 02 2025 10:55:00 GMT+0100 (British Summer Time)	completed	1	2	Vaccination
7	Sun Jul 20 2025 14:35:00 GMT+0100 (British Summer Time)	completed	1	1	Routine blood test
8	Tue Aug 05 2025 09:45:00 GMT+0100 (British Summer Time)	completed	1	2	Migraine
9	Mon Sep 01 2025 16:10:00 GMT+0100 (British Summer Time)	completed	1	1	Back pain
10	Tue Sep 30 2025 11:25:00 GMT+0100 (British Summer Time)	completed	1	2	Allergy flare
15	Sun Dec 07 2025 10:20:00 GMT+0000 (Greenwich Mean Time)	completed	1	1	Flu symptoms
17	Mon Dec 08 2025 14:45:00 GMT+0000 (Greenwich Mean Time)	completed	1	2	Follow-up call

Previous 1 Next

Status filter.

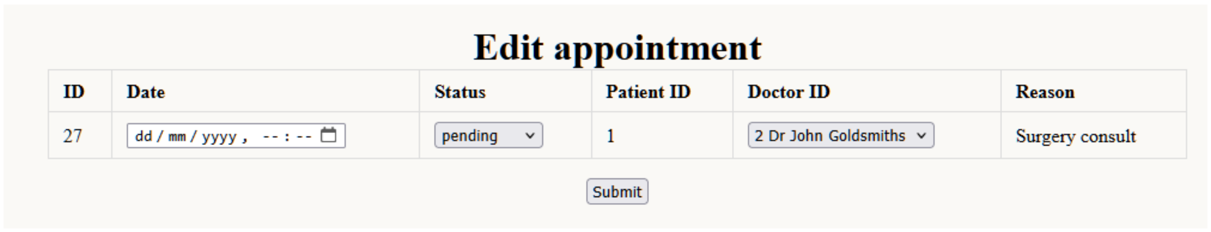

List of appointments

Date from: dd / mm / yyyy   Date to: dd / mm / yyyy   Status: Show unassigned: ☑
Apply filters

ID	Date	Status	Patient ID	Doctor ID	Reason
3		pending	14		Hi my name is king and I am the king of the world! I have a head.
6	Sun Nov 10 2024 13:30:00 GMT+0000 (Greenwich Mean Time)	cancelled	1		I am having trouble sleeping
13	Sun Nov 10 2024 13:30:00 GMT+0000 (Greenwich Mean Time)	cancelled	1		Annual checkup
18	Tue Dec 16 2025 12:00:00 GMT+0000 (Greenwich Mean Time)	pending	1		Physio session
19	Thu Dec 18 2025 15:10:00 GMT+0000 (Greenwich Mean Time)	pending	1		Consultation
22	Sun Jan 11 2026 09:10:00 GMT+0000 (Greenwich Mean Time)	cancelled	1		MRI Scan
23	Mon Jan 12 2026 09:50:00 GMT+0000 (Greenwich Mean Time)	pending	1		Routine check up
25	Fri Jan 16 2026 12:20:00 GMT+0000 (Greenwich Mean Time)	cancelled	1		Allergy test
28	Thu Mar 05 2026 16:55:00 GMT+0000 (Greenwich Mean Time)	pending	1		Vaccination booster
1		cancelled	1		I am feeling quite sick.

Unassigned filter (rows without a doctor).

Appointment Editing

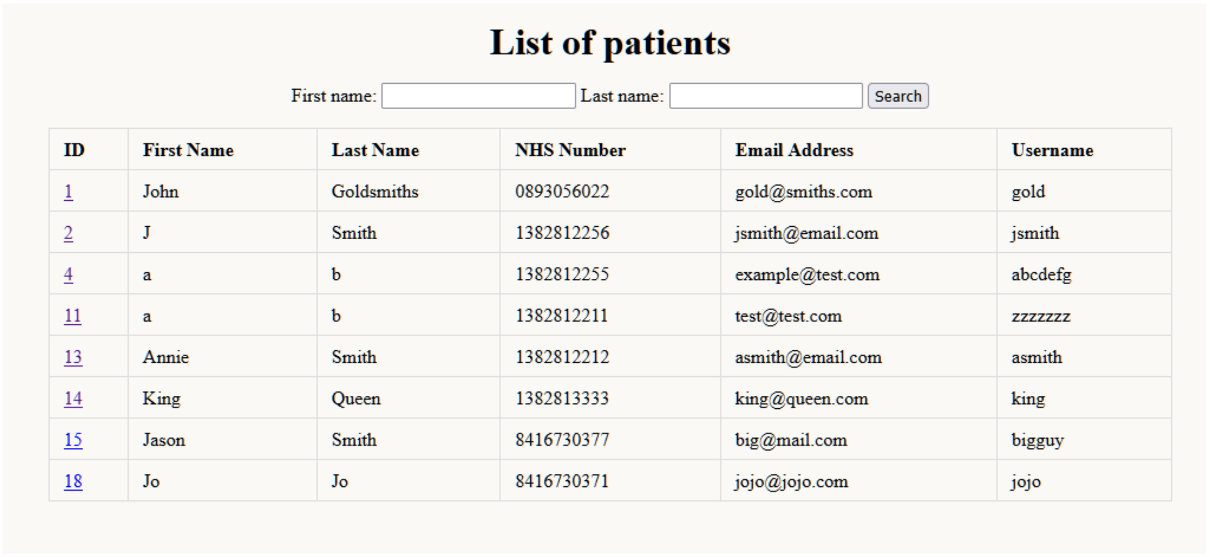

Edit appointment

ID	Date	Status	Patient ID	Doctor ID	Reason
27	dd / mm / yyyy , -- : --	pending	1	2 Dr John Goldsmiths	Surgery consult

Submit

Clicking on an appointment ID on the appointments table takes you to the individual page for that appointment, where its details can be edited. This can also be accessed through the patients table (discussed below). Each record has their own page and can be accessed directly by entering the url: (base domain)/doctors/appointments/*id here*. The date, status and doctor can be edited but not the patient ID or reason as they are irrevocably linked to an patients specific request, this allows for an audit trail.

Patients Table

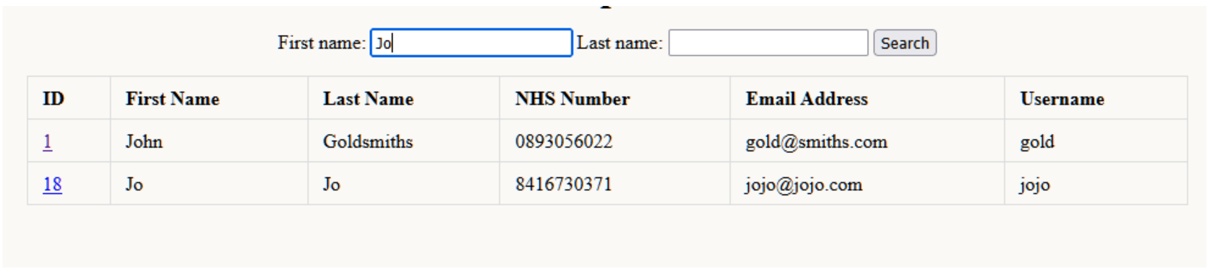


List of patients

First name: Last name: Search

ID	First Name	Last Name	NHS Number	Email Address	Username
1	John	Goldsmiths	0893056022	gold@smiths.com	gold
2	J	Smith	1382812256	jsmith@email.com	jsmith
4	a	b	1382812255	example@test.com	abcdefg
11	a	b	1382812211	test@test.com	zzzzzzz
13	Annie	Smith	1382812212	asmith@email.com	asmith
14	King	Queen	1382813333	king@queen.com	king
15	Jason	Smith	8416730377	big@mail.com	bigguy
18	Jo	Jo	8416730371	jojo@jojo.com	jojo

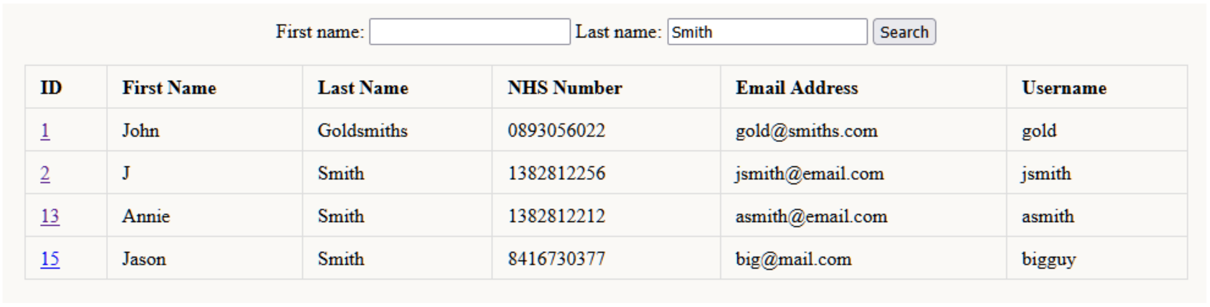
Accessed through the admin navbar, this table allows users to view a list of all patients registered to the practice. Users can search this table to find a specific patient using either their first or last name or both.



First name: Jo Last name: Search

ID	First Name	Last Name	NHS Number	Email Address	Username
1	John	Goldsmiths	0893056022	gold@smiths.com	gold
18	Jo	Jo	8416730371	jojo@jojo.com	jojo

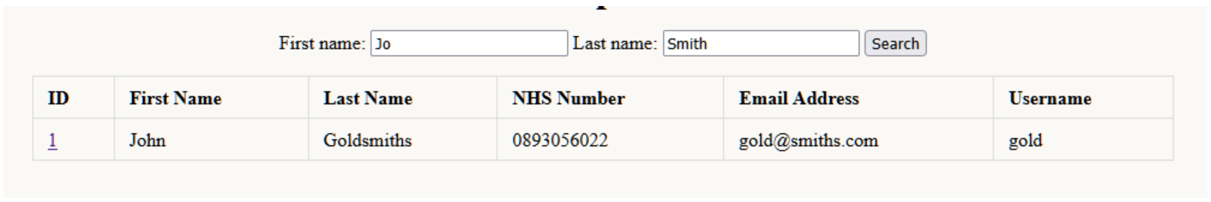
Searching first name only.



First name: Last name: Smith Search

ID	First Name	Last Name	NHS Number	Email Address	Username
1	John	Goldsmiths	0893056022	gold@smiths.com	gold
2	J	Smith	1382812256	jsmith@email.com	jsmith
13	Annie	Smith	1382812212	asmith@email.com	asmith
15	Jason	Smith	8416730377	big@mail.com	bigguy

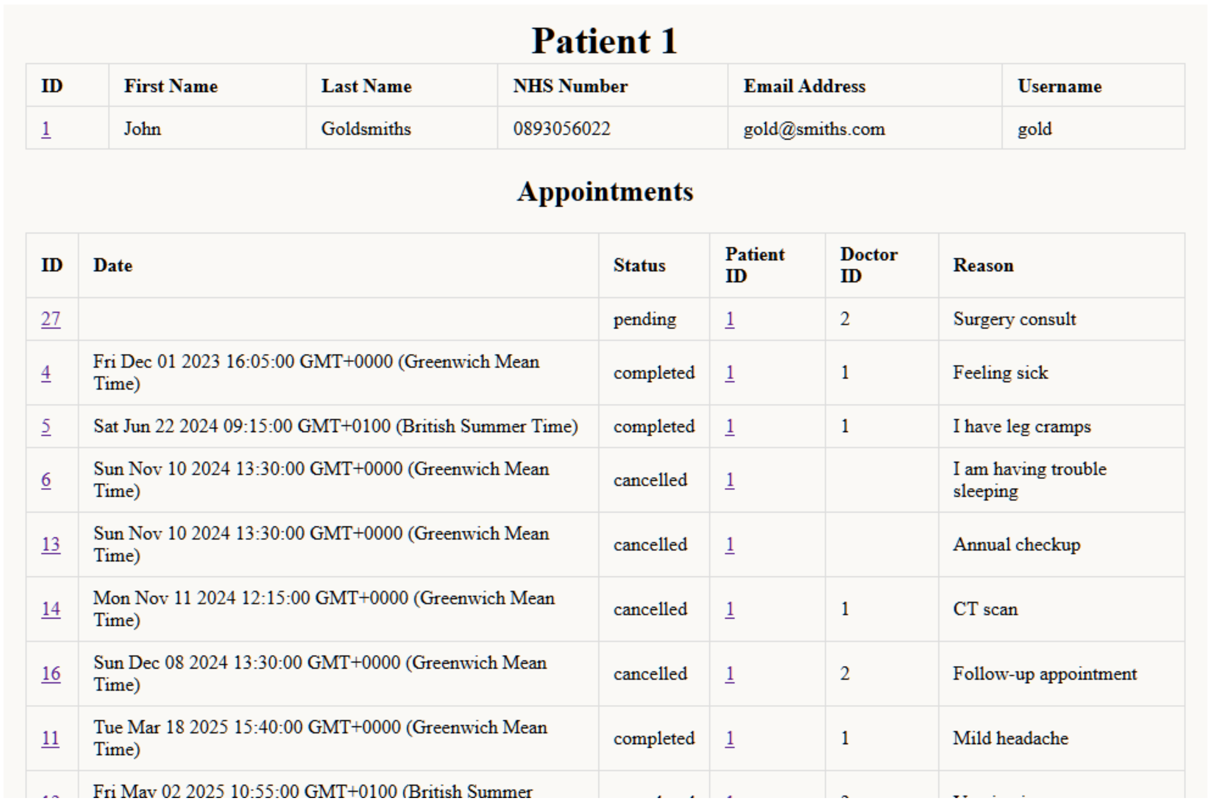
Searching last name only.


First name: Jo Last name: Smith Search

ID	First Name	Last Name	NHS Number	Email Address	Username
1	John	Goldsmiths	0893056022	gold@smiths.com	gold

Searching first and last name.

Individual Patient Record


Patient 1

ID	First Name	Last Name	NHS Number	Email Address	Username
1	John	Goldsmiths	0893056022	gold@smiths.com	gold

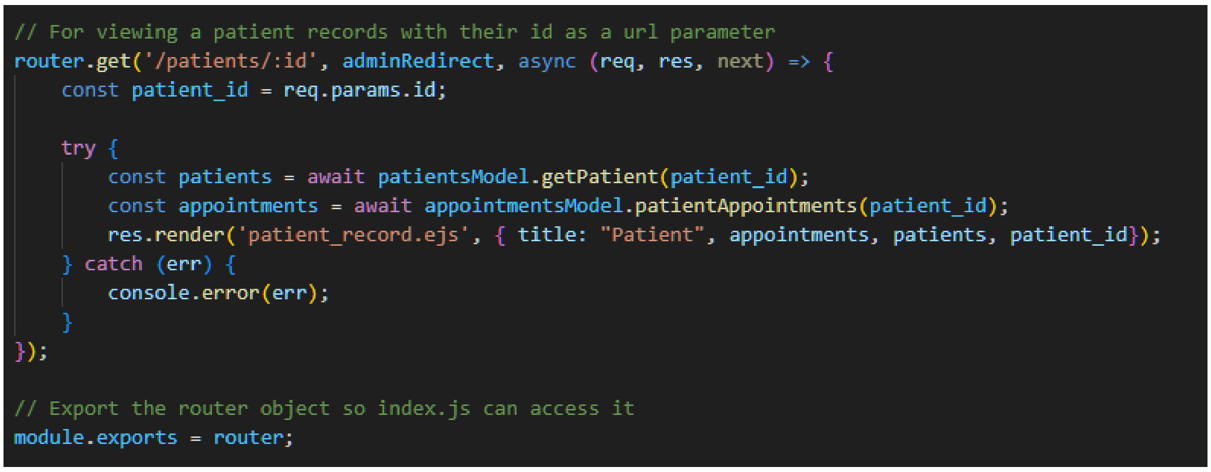
Appointments

ID	Date	Status	Patient ID	Doctor ID	Reason
27		pending	1	2	Surgery consult
4	Fri Dec 01 2023 16:05:00 GMT+0000 (Greenwich Mean Time)	completed	1	1	Feeling sick
5	Sat Jun 22 2024 09:15:00 GMT+0100 (British Summer Time)	completed	1	1	I have leg cramps
6	Sun Nov 10 2024 13:30:00 GMT+0000 (Greenwich Mean Time)	cancelled	1		I am having trouble sleeping
13	Sun Nov 10 2024 13:30:00 GMT+0000 (Greenwich Mean Time)	cancelled	1		Annual checkup
14	Mon Nov 11 2024 12:15:00 GMT+0000 (Greenwich Mean Time)	cancelled	1	1	CT scan
16	Sun Dec 08 2024 13:30:00 GMT+0000 (Greenwich Mean Time)	cancelled	1	2	Follow-up appointment
11	Tue Mar 18 2025 15:40:00 GMT+0000 (Greenwich Mean Time)	completed	1	1	Mild headache
	Fri May 02 2025 10:55:00 GMT+0100 (British Summer				

Similar to the appointment ID, users can click on the patient ID on the patients table, appointments table or dashboard to view an individual patient record. This includes their account details along with a list of appointments linked to them. It can also be accessed through (base domain)/doctors/patients/*id here*.

Advanced Techniques

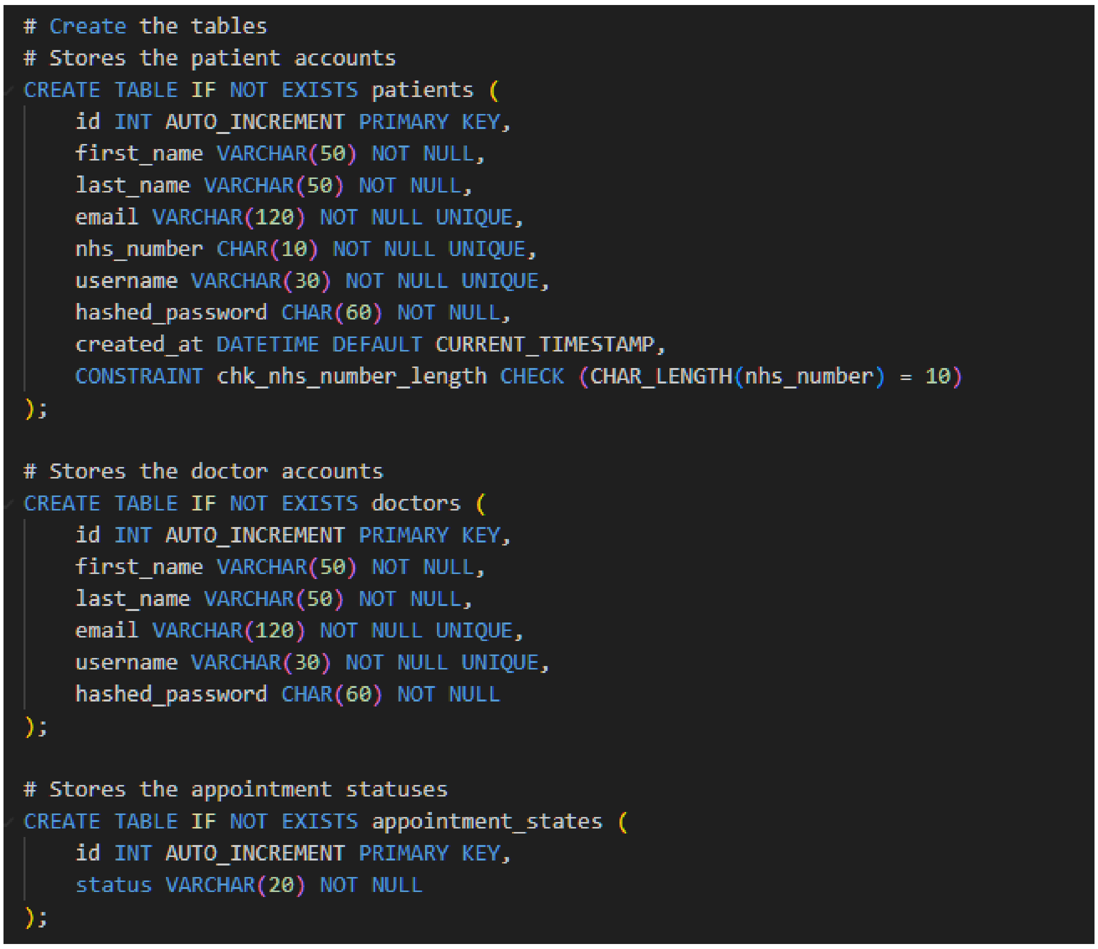
Dynamic Routing



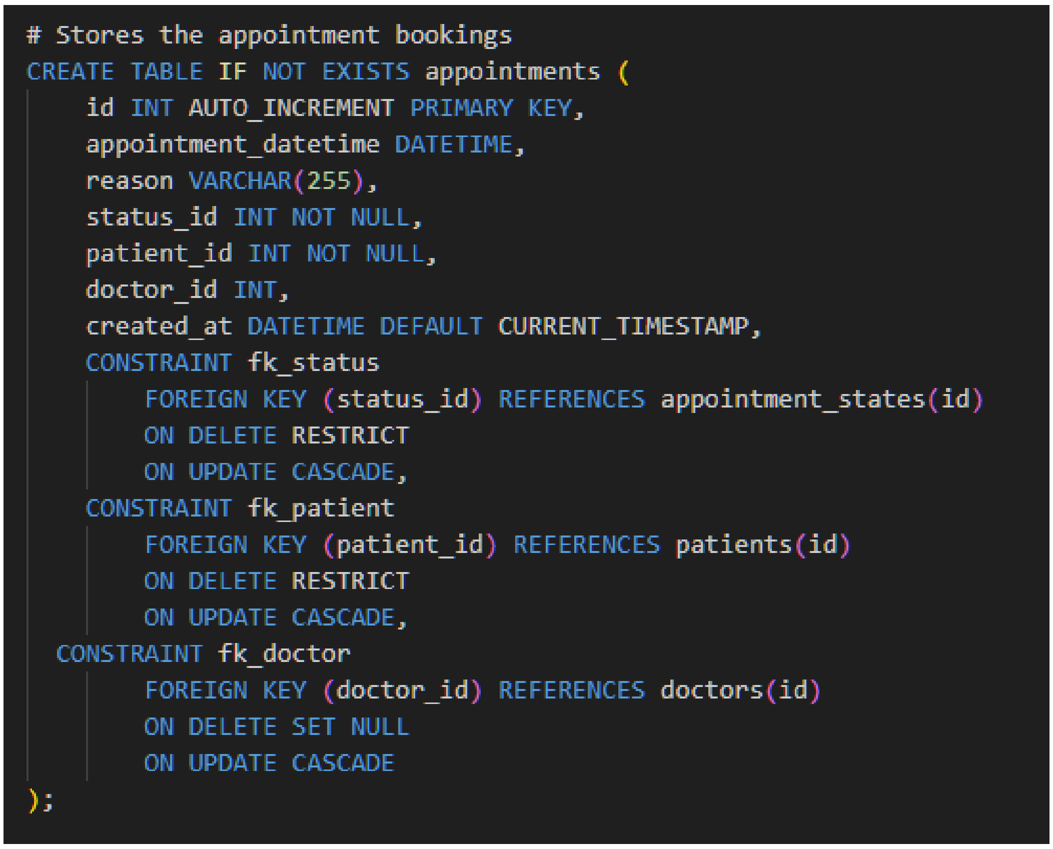
```javascript
// For viewing a patient records with their id as a url parameter
router.get('/patients/:id', adminRedirect, async (req, res, next) => {
    const patient_id = req.params.id;

    try {
        const patients = await patientsModel.getPatient(patient_id);
        const appointments = await appointmentsModel.patientAppointments(patient_id);
        res.render('patient_record.ejs', { title: "Patient", appointments, patients, patient_id});
    } catch (err) {
        console.error(err);
    }
});

// Export the router object so index.js can access it
module.exports = router;
```

The web app is automatically populated with a page for each record in the appointments and patients table. This reduces the need to manually create a page for every individual record and complies with DRY principles.

Relational Databases and Advance Schema



```sql
# Create the tables
# Stores the patient accounts
CREATE TABLE IF NOT EXISTS patients (
    id INT AUTO_INCREMENT PRIMARY KEY,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    email VARCHAR(120) NOT NULL UNIQUE,
    nhs_number CHAR(10) NOT NULL UNIQUE,
    username VARCHAR(30) NOT NULL UNIQUE,
    hashed_password CHAR(60) NOT NULL,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    CONSTRAINT chk_nhs_number_length CHECK (CHAR_LENGTH(nhs_number) = 10)
);

# Stores the doctor accounts
CREATE TABLE IF NOT EXISTS doctors (
    id INT AUTO_INCREMENT PRIMARY KEY,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    email VARCHAR(120) NOT NULL UNIQUE,
    username VARCHAR(30) NOT NULL UNIQUE,
    hashed_password CHAR(60) NOT NULL
);

# Stores the appointment statuses
CREATE TABLE IF NOT EXISTS appointment_states (
    id INT AUTO_INCREMENT PRIMARY KEY,
    status VARCHAR(20) NOT NULL
);
```

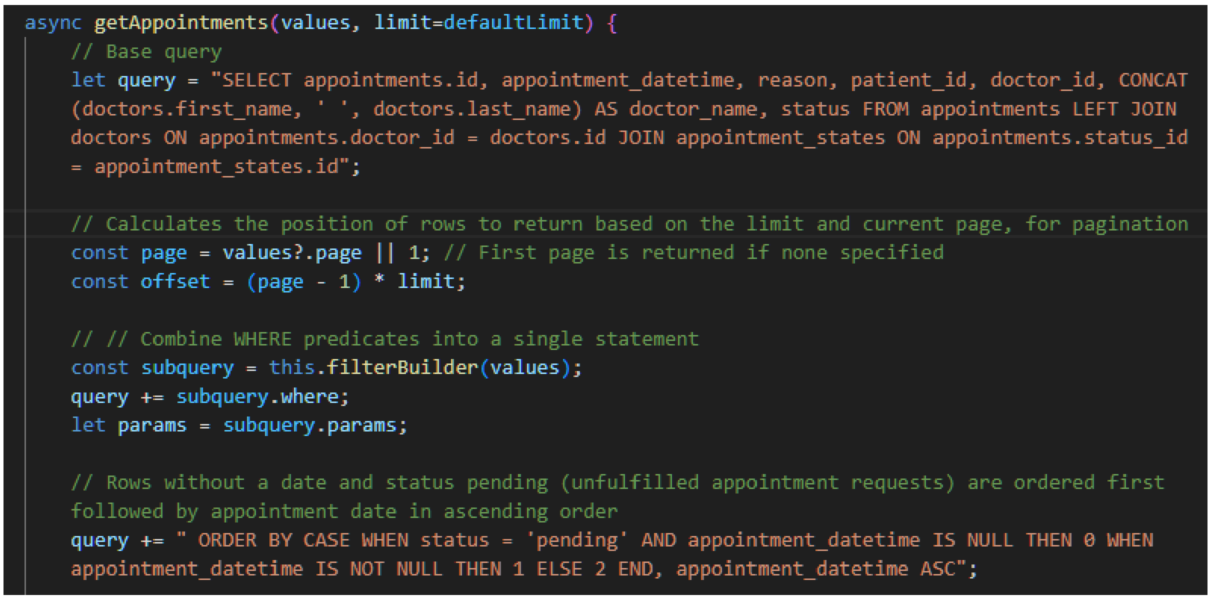


```sql
# Stores the appointment bookings
CREATE TABLE IF NOT EXISTS appointments (
    id INT AUTO_INCREMENT PRIMARY KEY,
    appointment_datetime DATETIME,
    reason VARCHAR(255),
    status_id INT NOT NULL,
    patient_id INT NOT NULL,
    doctor_id INT,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    CONSTRAINT fk_status
        FOREIGN KEY (status_id) REFERENCES appointment_states(id)
        ON DELETE RESTRICT
        ON UPDATE CASCADE,
    CONSTRAINT fk_patient
        FOREIGN KEY (patient_id) REFERENCES patients(id)
        ON DELETE RESTRICT
        ON UPDATE CASCADE,
  CONSTRAINT fk_doctor
        FOREIGN KEY (doctor_id) REFERENCES doctors(id)
        ON DELETE SET NULL
        ON UPDATE CASCADE
);
```

The database schema includes techniques that go beyond the basics of database design.

- Foreign keys for table joins (E.g. status ID, doctor ID and patient ID in appointments),
- NHS number check constraint that only accepts values of a specific length.
- Parent table (appointments) with a separate table (appointment states) for storing static values.
- Foreign key constraints.

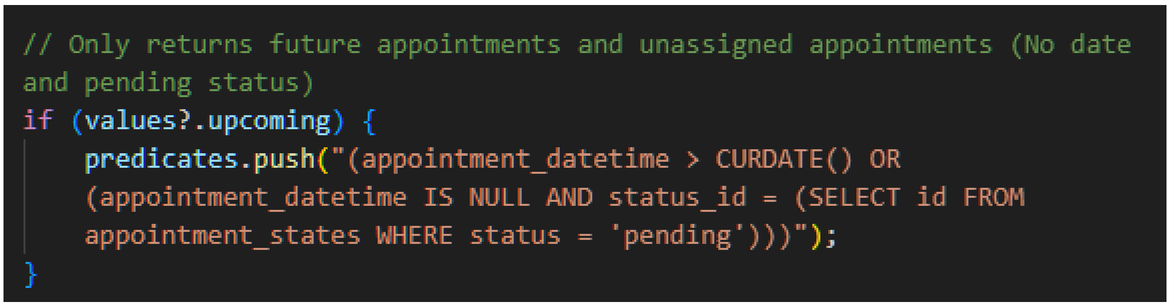
Advance Database Techniques



```javascript
async getAppointments(values, limit=defaultLimit) {
    // Base query
    let query = "SELECT appointments.id, appointment_datetime, reason, patient_id, doctor_id, CONCAT
    (doctors.first_name, ' ', doctors.last_name) AS doctor_name, status FROM appointments LEFT JOIN
    doctors ON appointments.doctor_id = doctors.id JOIN appointment_states ON appointments.status_id
    = appointment_states.id";

    // Calculates the position of rows to return based on the limit and current page, for pagination
    const page = values?.page || 1; // First page is returned if none specified
    const offset = (page - 1) * limit;

    // // Combine WHERE predicates into a single statement
    const subquery = this.filterBuilder(values);
    query += subquery.where;
    let params = subquery.params;

    // Rows without a date and status pending (unfulfilled appointment requests) are ordered first
    followed by appointment date in ascending order
    query += " ORDER BY CASE WHEN status = 'pending' AND appointment_datetime IS NULL THEN 0 WHEN
    appointment_datetime IS NOT NULL THEN 1 ELSE 2 END, appointment_datetime ASC";
```

Carefully constructed queries are constructed, so that all the necessary access requirements are fulfilled while including additional data from other tables.

- Table joins are utilised to gain access to variables from another table, with join type considered. For example, left join is used for doctors ID as not all appointment rows will have a doctors ID, this ensures rows with null values are kept in the resulting join.
- CONCAT combines two columns into one and the new column is given a name.
- CASE WHEN is used to allow a custom ordering specific to the app requirements.

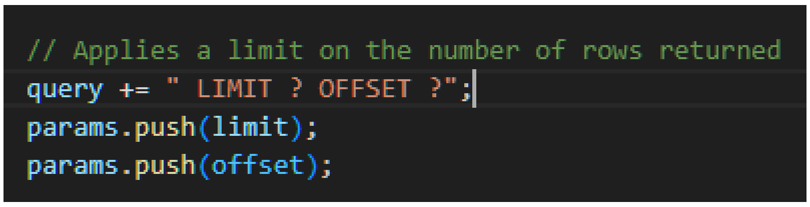


```
// Only returns future appointments and unassigned appointments (No date
and pending status)
if (values?.upcoming) {
    predicates.push("(appointment_datetime > CURDATE() OR
    (appointment_datetime IS NULL AND status_id = (SELECT id FROM
    appointment_states WHERE status = 'pending')))");
}
```

- WHERE conditions are used for filtering results. A key example of this is the ‘upcoming’ filter, it uses the CURDATE function to retrieve the today’s date so that it can be compared against the appointments date. This ensures that only appointment dates in the future are selected. Along with this, appointments without a date and with a ‘pending’ status are also selected, as these represent new requests by patients.

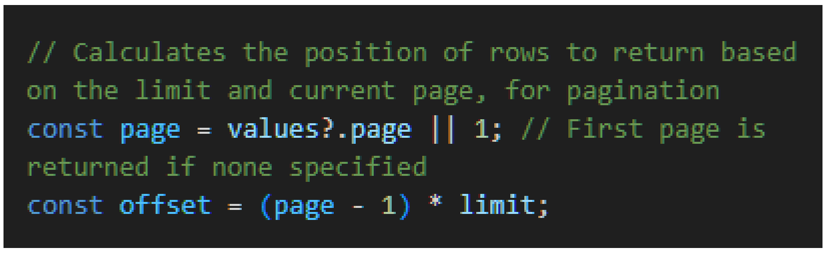
Pagination

Appointments table is paginated, only a subset of rows is retrieved and displayed at a time.



```
// Applies a limit on the number of rows returned
query += " LIMIT ? OFFSET ?";
params.push(limit);
params.push(offset);
```

Using LIMIT and OFFSET conditions.



```
// Calculates the position of rows to return based
on the limit and current page, for pagination
const page = values?.page || 1; // First page is
returned if none specified
const offset = (page - 1) * limit;
```

LIMIT and OFFSET are calculated using the current page variable.

```
// Returns the total amount of rows with filters
async rowCount(values) {
    let query = "SELECT COUNT(*) AS total FROM
    appointments";
    const subquery = this.filterBuilder(values);
    query += subquery.where;
    let params = subquery.params;
    const [result] = await db.query(query, params);
    return result[0].total;
},
```

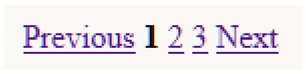
A separate query for the row count to reduce complexity of the main query and to not include an unnecessary 'total' column on every row.

```
try {
    const totalRows = await appointmentsModel.rowCount(filters);
    const totalPages = Math.ceil(totalRows / limit);
    const page = Math.min(Math.max(Number(filters?.page), 1), totalPages);
    filters.page = page;
```

The number of pages is then calculated using the row count. Current page is also constrained within a range of 1 to the total number of pages so that the user cannot underflow or overflow the page number.

```
<!-- Pagination controls including page numbers -->
 <div class="table-wrapper">
<%- include('./partials/appointmentsTable') %>
<div class="table-links">
<a href="<%= process.env.HEALTH_BASE_PATH %>/doctors/appointments?<%= new
URLSearchParams({...filters, page: Math.min(Math.max(Number(page) -1, 1),
totalPages) }) %>">Previous</a>
<% for (let i = 1; i < totalPages+1; i++) { %>
  <% if (i === Number(filters.page)) { %>
    <span style="font-weight: bold;"><%= i %></span>
  <% } else { %>
    <a href="<%= process.env.HEALTH_BASE_PATH %>/doctors/appointments?<%= new
    URLSearchParams({ ...filters, page: i }) %>"><%= i %></a>
  <% } %>
<% } %>
<a href="<%= process.env.HEALTH_BASE_PATH %>/doctors/appointments?<%= new
URLSearchParams({...filters, page: Math.min(Math.max(Number(page) +1, 1),
totalPages) }) %>">Next</a>
  </div>
</div>
```

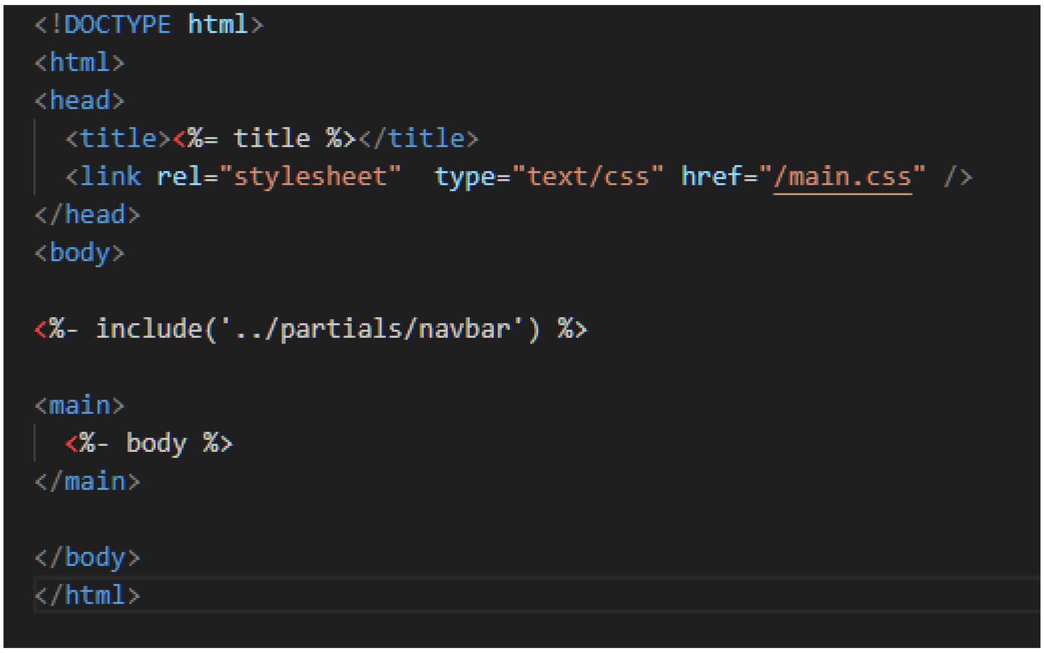
A for-loop then populates the appointments table view with links to each page, as well as next and previous links.


Previous **1** 2 3 Next

Result: Number of pages displayed. Current page is bold text instead of a hyperlink.

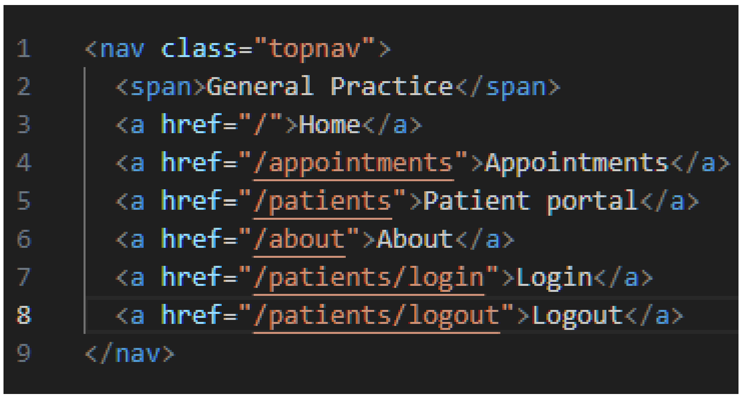
Layouts and Render Partial

This reduces repetitive HTML and produces building blocks that can be repeated throughout.



```html
<!DOCTYPE html>
<html>
<head>
  <title><%= title %></title>
  <link rel="stylesheet"  type="text/css" href="/main.css" />
</head>
<body>

<%- include('../partials/navbar') %>

<main>
  <%- body %>
</main>

</body>
</html>
```

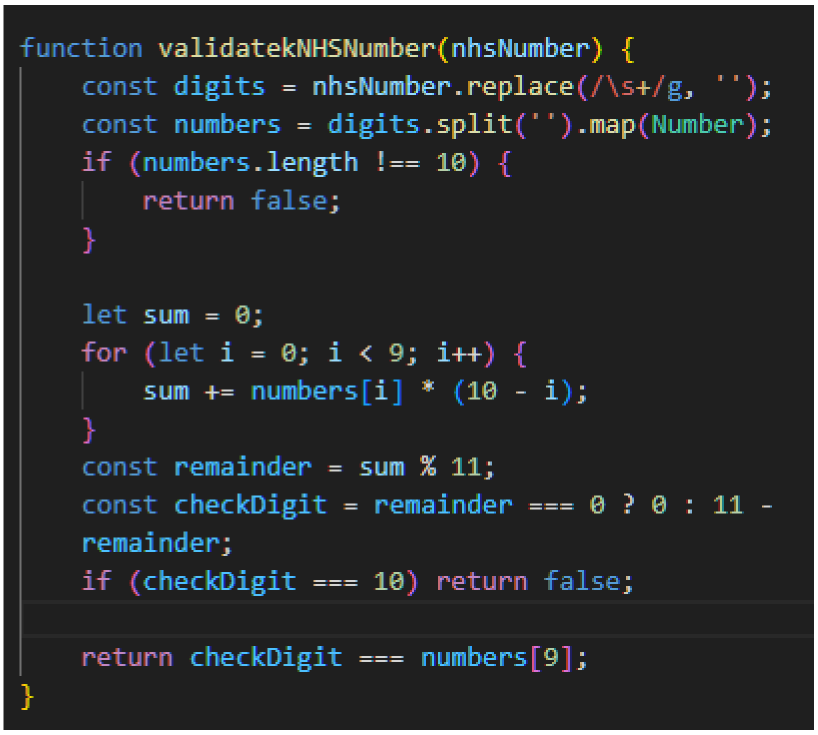
Layout



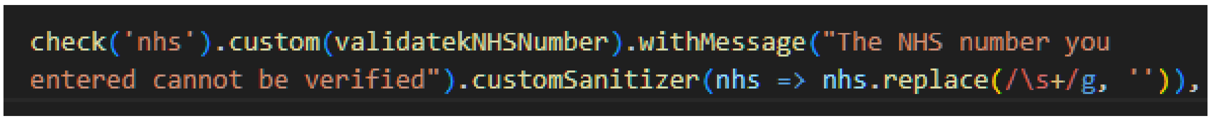
```html
<nav class="topnav">
  <span>General Practice</span>
  <a href="/">Home</a>
  <a href="/appointments">Appointments</a>
  <a href="/patients">Patient portal</a>
  <a href="/about">About</a>
  <a href="/patients/login">Login</a>
  <a href="/patients/logout">Logout</a>
</nav>
```

Partial

Custom Validation



```
function validatekNHSNumber(nhsNumber) {
    const digits = nhsNumber.replace(/\s+/g, '');
    const numbers = digits.split('').map(Number);
    if (numbers.length !== 10) {
        return false;
    }

    let sum = 0;
    for (let i = 0; i < 9; i++) {
        sum += numbers[i] * (10 - i);
    }
    const remainder = sum % 11;
    const checkDigit = remainder === 0 ? 0 : 11 -
    remainder;
    if (checkDigit === 10) return false;

    return checkDigit === numbers[9];
}
```



```
check('nhs').custom(validatekNHSNumber).withMessage("The NHS number you
entered cannot be verified").customSanitizer(nhs => nhs.replace(/\s+/g, '')),
```

A validation function for NHS numbers, an attribute that has unique validation rules. The function is then applied to express validation

AI Declaration

AI was used minimally, but it was for the following:

- Assistance with styling and aesthetic design.
- Variable and file naming.
- Generating test data.