

HTML Template Files with Detailed Comments

1. index.html (Home Page)

```
<!DOCTYPE html>

<!-- HTML5 document type declaration --&gt;

&lt;html xmlns:th='http://www.thymeleaf.org'&gt;

&lt;!--
    xmlns:th='http://www.thymeleaf.org' - Thymeleaf namespace declaration

    This allows us to use Thymeleaf expressions and attributes in the HTML
--&gt;

&lt;head&gt;
    &lt;title&gt;Home&lt;/title&gt;

    &lt;!-- Page title that appears in browser tab --&gt;

    &lt;!-- Link to external CSS stylesheet --&gt;
    &lt;link rel='stylesheet' href='/css/style.css' /&gt;

    &lt;!--
        rel='stylesheet' - Specifies the relationship as a stylesheet
        href='/css/style.css' - Path to the CSS file (static resources)
--&gt;

    &lt;!-- Inline CSS styles for the home page --&gt;
    &lt;style&gt;
        /*
            Inline CSS - styles defined within the HTML document
            This is used for page-specific styling that doesn't need to be in external
            CSS
        */
    </pre>
```

```
/* Body styling - sets overall page appearance */

body {
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    /*

        Modern font stack - tries Segoe UI first, then falls back to other fonts

        This ensures consistent typography across different operating systems

    */
    background-color: #f4f6f8; /* Light gray background color */
    margin: 0;             /* Remove default browser margins */
    padding: 20px;          /* Add padding around content */
    color: #333;            /* Dark gray text color for good readability */
    flex-direction: column; /* Vertical layout for flexbox children */
}

/* Container for main content */

.ing {
    flex-direction: column; /* Vertical layout for content */
    justify-items: center; /* Center content horizontally */
}

/* Main heading styling */

h1 {
    color: #2c3e50;      /* Dark blue color for heading */
    font-size: 28px;       /* Large font size for prominence */
    margin-bottom: 10px;   /* Space below heading */
    border-bottom: 2px solid #3498db; /* Blue underline for visual appeal */
    padding-bottom: 5px;   /* Space above underline */
}
```

```
/* Image container styling */

#imd {
    padding: 20px;      /* Padding around image */
    box-shadow: inset;  /* Inset shadow effect for depth */
}

/* Hospital image styling */

#img1 {
    border-radius: 20px; /* Rounded corners for modern look */
}

/* Footer styling */

footer {
    position: fixed;    /* Fixed position at bottom of viewport */
    bottom: 0;           /* Stick to bottom of screen */
    width: 100%;         /* Full width of viewport */
    text-align: center;  /* Center text horizontally */
    color: rgb(13, 8, 3); /* Dark brown color for footer text */
}

</style>

</head>

<body>

<!--

    Include navigation bar fragment

    th:insert='fragments/navbar :: fragment' - Thymeleaf fragment inclusion

    This inserts the navbar.html fragment into this page


```

```
-->

<div th:insert='fragments/navbar :: fragment'></div>

<!-- Main content container -->

<div class="ing">

    <!-- Main page heading -->

    <h1>Patient Appointment Management System</h1>

    <!--

        Static heading - no Thymeleaf expressions needed

        This is the main title of the application

    -->

    <!-- Image container -->

    <div id="imd">

        <!--

            Hospital image with rounded corners

            src - URL to the image (external image from drlogy.com)

            alt - Alternative text for accessibility

        -->

        

    </div>

    <!-- Footer with copyright information -->

    <footer>THIS SITE BELONGS TO ©ISG25JS001-POD-5, 2025.</footer>

    <!--

        © - HTML entity for copyright symbol

        Static footer with project information

    -->
```

```
-->

</div>

</body>

</html>
```

2. fragments/navbar.html (Navigation Bar) -

```
<!--
```

Navigation bar fragment

This is a reusable component that can be included in any page

th:fragment="fragment" - Defines this as a reusable Thymeleaf fragment

```
-->
```

```
<div
```

style="background-color: #eee; padding: 10px; font-family: Arial, sans-serif; font-size: 16px;"

th:fragment="fragment">

```
<!--
```

Inline styles for the navigation bar:

- background-color: #eee - Light gray background

- padding: 10px - Space inside the navbar

- font-family: Arial, sans-serif - Simple, readable font

- font-size: 16px - Standard font size for navigation

```
-->
```

```
<!-- Home link -->
```

Home

```
<!--
```

href="/" - Links to the root URL (home page)

style attributes:

- margin: 0 5px - Space around the link
- text-decoration: none - Removes underline
- color: #000 - Black text color

-->

<!-- Patient registration link -->

| Patient Register

<!--

- | - Separator character between navigation items
- href="/patients/register" - Links to patient registration page
- Same styling as home link for consistency

-->

<!-- Patient login link -->

| Patient Login

<!--

- href="/patients/login" - Links to patient login page
- Consistent styling with other navigation links

-->

<!-- Doctors list link -->

| Doctors

<!--

- href="/doctors" - Links to doctors listing page
- Allows users to browse available doctors

-->

```
<!-- Book appointment link -->
| <a href="/appointments/book" style="margin: 0 5px; text-decoration: none; color: #000;">Book Appointment</a>
<!--
    href="/appointments/book" - Links to appointment booking page
    Main functionality for patients to book appointments
-->

<!-- Admin login link (right-aligned) -->
| <a href="/admin/login" style="margin: 0 5px; text-decoration: none; color: #000; float: right;">Admin</a>
<!--
    href="/admin/login" - Links to admin login page
    float: right - Positions this link on the right side of the navbar
    Double semicolon (;;) appears to be a typo but doesn't break functionality
-->
</div>

<!-- Horizontal rule separator -->
<hr />
<!--
    <hr /> - Creates a horizontal line to separate navigation from page content
    Self-closing tag (XHTML style)
-->
3. patient-login.html (Patient Login Form)

<!DOCTYPE html>
<!-- HTML5 document type declaration -->
<html xmlns:th='http://www.thymeleaf.org'>
```

```
<!--  
    xmlns:th='http://www.thymeleaf.org' - Thymeleaf namespace  
    Enables Thymeleaf expressions and attributes  
-->  
  
<head>  
    <title>Login</title>  
    <!-- Page title for browser tab -->  
  
    <!-- Link to external CSS stylesheet -->  
    <link rel='stylesheet' href='/css/style.css' />  
    <!--  
        External stylesheet for consistent styling across the application  
        Static resource served from /css/ directory  
-->  
  
    <!-- Inline CSS styles for the login page -->  
    <style>  
        /*  
            Page-specific styles that override or supplement external CSS  
            These styles are specific to the login page layout  
        */  
  
        /* Body styling */  
        body {  
            font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;  
            /*  
                Modern font stack for consistent typography  
                Falls back gracefully across different operating systems  
            */  
        }  
    </style>
```

```
*/  
  
background: #f4f6f8; /* Light gray background for subtle contrast */  
margin: 0; /* Remove default browser margins */  
padding: 20px; /* Add padding around page content */  
}  
  
/* Footer styling */  
  
footer {  
position: fixed; /* Fixed position at bottom of viewport */  
bottom: 0; /* Stick to bottom of screen */  
width: 100%; /* Full width of viewport */  
text-align: center; /* Center text horizontally */  
color: rgb(13, 8, 3); /* Dark brown color for footer text */  
}  
  
/* Heading styling */  
  
h2 {  
text-align: center; /* Center the heading horizontally */  
color: #333; /* Dark gray color for good readability */  
margin-top: 40px; /* Space above heading for visual separation */  
}  
  
/* Form styling */  
  
form {  
max-width: 400px; /* Limit form width for better UX */  
margin: 30px auto; /* Center form horizontally with top margin */  
background: #fff; /* White background for form */  
padding: 25px; /* Padding inside form for breathing room */
```

```
border-radius: 8px;      /* Rounded corners for modern look */
box-shadow: 0 0 10px rgba(0,0,0,0.1);
/*
Subtle shadow effect:
- 0 0 - No horizontal or vertical offset
- 10px - Blur radius
- rgba(0,0,0,0.1) - Semi-transparent black (10% opacity)
*/
}

/* Input field styling */
input[type='email'],
input[type='password']{
    width: 100%;          /* Full width of container */
    padding: 10px;         /* Padding inside input fields */
    margin: 10px 0 20px 0; /* Margin: top right bottom left */
    border: 1px solid #ccc; /* Light gray border */
    border-radius: 4px;    /* Slightly rounded corners */
    font-size: 14px;       /* Standard font size for inputs */
}

/* Button styling */
button{
    width: 100%;          /* Full width of container */
    padding: 10px;         /* Padding inside button */
    background-color: #007bff; /* Bootstrap blue color */
    color: white;          /* White text for contrast */
    border: none;          /* Remove default border */
}
```

```
border-radius: 4px;      /* Rounded corners */
font-size: 16px;        /* Larger font for button text */
cursor: pointer;        /* Pointer cursor on hover */

}

/* Button hover effect */

button:hover {
    background-color: #0056b3; /* Darker blue on hover for feedback */
}

/* Error message styling */

p[th\:text='${error}'] {
    /*
        CSS selector for paragraph with Thymeleaf error text
        th\:text - Escaped Thymeleaf attribute in CSS selector
    */
    color: red;          /* Red text for error messages */
    text-align: center;   /* Center error message */
    margin-top: 10px;     /* Space above error message */
}

</style>

</head>

<body>
    <!-- Include navigation bar fragment -->
    <div th:insert='fragments/navbar :: fragment'></div>
    <!--
        th:insert - Thymeleaf directive to insert a fragment
    -->

```

'fragments/navbar :: fragment' - Path to fragment and fragment name

This includes the navigation bar in this page

-->

<!-- Page heading -->

<h2>Patient Login</h2>

<!-- Static heading for the login page -->

<!-- Login form -->

<!--

th:action='@{/patients/login}' - Thymeleaf URL expression for form action

@{} - URL expression that generates proper URLs

method='post' - HTTP POST method for form submission

-->

<form th:action='@{/patients/login}' method='post'>

<!-- Email input field -->

Email: <input type='email' name='email' required />

<!--

type='email' - HTML5 email input type with built-in validation

name='email' - Parameter name sent to server

required - HTML5 validation attribute (field must be filled)

 - Line break after input

-->

<!-- Password input field -->

Password: <input type='password' name='password' required />

<!--

type='password' - Password input type (text is hidden)

```
    name='password' - Parameter name sent to server
    required - HTML5 validation attribute

-->

<!-- Submit button -->
<button type='submit'>Login</button>
<!--
    type='submit' - Submits the form when clicked
    Text content is the button label
-->
</form>

<!-- Error message display -->
<!--
    th:text='${error}' - Thymeleaf expression to display error message
    ${error} - Variable from the model (set by controller)
    Only displays if error variable exists and is not null
-->
<p th:text='${error}'></p>

<!-- Footer -->
<footer>THIS SITE BELONGS TO ©ISG25JS001-POD-5, 2025.</footer>
<!--
    © - HTML entity for copyright symbol
    Static footer with project information
-->
</body>
</html>
```

4. patient-register.html (Patient Registration Form)

```
<!DOCTYPE html>

<!-- HTML5 document type declaration -->

<html xmlns:th='http://www.thymeleaf.org'>

<!--
    xmlns:th='http://www.thymeleaf.org' - Thymeleaf namespace
    Enables Thymeleaf expressions and form binding
-->

<head>
    <title>Register</title>
    <!-- Page title for browser tab -->

    <!-- Link to external CSS stylesheet -->
    <link rel='stylesheet' href='/css/style.css' />
    <!-- External stylesheet for consistent application styling -->

    <!-- Inline CSS styles for the registration page -->
    <style>
        /* Page-specific styles for registration form */

        /* Body styling */
        body {
            font-family: Arial, sans-serif; /* Simple, readable font */
            background-color: #f4f6f8; /* Light gray background */
            margin: 0; /* Remove default browser margins */
            padding: 20px; /* Add padding around content */
        }
    
```

```
/* Footer styling */

footer {
    position: fixed;          /* Fixed position at bottom */
    bottom: 0;                 /* Stick to bottom of viewport */
    width: 100%;               /* Full width of viewport */
    text-align: center;         /* Center text horizontally */
    color: rgb(13, 8, 3);      /* Dark brown color */
}

/* Heading styling */

h2 {
    color: #2c3e50;           /* Dark blue color */
    text-align: center;         /* Center the heading */
    margin-bottom: 30px;        /* Space below heading */
}

/* Form styling */

form {
    max-width: 500px;          /* Limit form width for better UX */
    margin: 0 auto;             /* Center form horizontally */
    background: #ffffff;        /* White background for form */
    padding: 30px;              /* Padding inside form */
    border-radius: 8px;          /* Rounded corners */
    box-shadow: 0 0 10px rgba(0,0,0,0.1);
    /* Subtle shadow for depth */
}

/* Input field styling for all input types */
```

```
input[type='text'],
input[type='email'],
input[type='date'],
input[type='password']{
    width: 100%;          /* Full width of container */
    padding: 10px;         /* Padding inside inputs */
    margin: 8px 0 16px 0;   /* Margin: top right bottom left */
    border: 1px solid #ccc;  /* Light gray border */
    border-radius: 4px;     /* Rounded corners */
    box-sizing: border-box; /* Include padding in width calculation */
}

/* Button styling */

button{
    background-color: #3498db;  /* Blue background */
    color: white;             /* White text */
    padding: 12px 20px;        /* Padding inside button */
    border: none;              /* Remove default border */
    border-radius: 4px;         /* Rounded corners */
    cursor: pointer;           /* Pointer cursor on hover */
    width: 100%;               /* Full width of container */
    font-size: 16px;            /* Standard font size */
}

/* Button hover effect */

button:hover{
    background-color: #2980b9;  /* Darker blue on hover */
}
```

```
</style>

</head>

<body>

    <!-- Include navigation bar fragment -->

    <div th:insert='fragments/navbar :: fragment'></div>

    <!--

        th:insert - Thymeleaf directive to include navbar fragment

        This provides consistent navigation across all pages

    -->

    <!-- Page heading -->

    <h2>Patient Register</h2>

    <!-- Static heading for registration page -->

    <!-- Registration form -->

    <!--

        th:action='@{/patients/register}' - Thymeleaf URL expression for form action

        method='post' - HTTP POST method for form submission

        th:object='${patient}' - Thymeleaf object binding for form data

        This binds the form to a Patient object from the model

    -->

    <form th:action='@{/patients/register}' method='post' th:object='${patient}'>

        <!-- Name input field -->

        <!--

            th:field='*{name}' - Thymeleaf field binding to patient.name

            *{} - Selection variable expression (refers to th:object)

            type='text' - Text input type

        -->

    
```

required - HTML5 validation attribute

-->

Name: <input type='text' th:field='*{name}' required />

<!-- Email input field -->

<!--

th:field='*{email}' - Binds to patient.email

type='email' - HTML5 email input with validation

required - Must be filled

-->

Email: <input type='email' th:field='*{email}' required />

<!-- Phone input field -->

<!--

th:field='*{phone}' - Binds to patient.phone

type='text' - Text input for phone number

No required attribute - phone is optional

-->

Phone: <input type='text' th:field='*{phone}' />

<!-- Address input field -->

<!--

th:field='*{address}' - Binds to patient.address

type='text' - Text input for address

Optional field

-->

Address: <input type='text' th:field='*{address}' />

```
<!-- Date of birth input field -->  
<!--  
    th:field='*{dob}' - Binds to patient.dob (LocalDate)  
    type='date' - HTML5 date picker  
    Optional field  
-->  
DOB: <input type='date' th:field='*{dob}' />
```

```
<!-- Password input field -->  
<!--  
    th:field='*{password}' - Binds to patient.password  
    type='password' - Password input (text hidden)  
    required - Must be filled  
-->  
Password: <input type='password' th:field='*{password}' required />
```

```
<!-- Submit button -->  
<button type='submit'>Register</button>  
<!--  
    type='submit' - Submits the form when clicked  
    Text content is the button label  
-->  
</form>
```

```
<!-- Footer -->  
<footer>THIS SITE BELONGS TO ©ISG25JS001-POD-5, 2025.</footer>  
<!--  
    © - HTML entity for copyright symbol
```

Static footer with project information

-->

```
<!-- JavaScript for date validation -->
```

```
<script>
```

```
//
```

```
// Client-side JavaScript for date validation
```

```
// This runs in the browser to enhance user experience
```

```
//
```

```
// Get the date of birth input field by ID
```

```
const dobInput = document.getElementById('dob');
```

```
//
```

```
// Note: This assumes the input has id='dob'
```

```
// The th:field attribute should generate this ID automatically
```

```
//
```

```
// Get today's date in YYYY-MM-DD format
```

```
const today = new Date().toISOString().split('T')[0];
```

```
//
```

```
// new Date() - Current date and time
```

```
// toISOString() - Converts to ISO string format (YYYY-MM-  
DDTHH:mm:ss.sssZ)
```

```
// split('T')[0] - Splits on 'T' and takes first part (date only)
```

```
//
```

```
// Set maximum date to today (prevents future dates)
```

```
dobInput.setAttribute('max', today);
```

```
//  
// setAttribute('max', today) - Sets HTML5 max attribute  
// This prevents users from selecting future dates  
// HTML5 date input will show validation error if future date is selected  
  
//  
</script>  
  
</body>  
  
</html>  
  
5. patient-dashboard.html (Patient Dashboard)  
  
<!DOCTYPE html>  
  
<!-- HTML5 document type declaration -->  
  
<html xmlns:th='http://www.thymeleaf.org'>  
  
<!--  
    xmlns:th='http://www.thymeleaf.org' - Thymeleaf namespace  
    Enables Thymeleaf expressions and conditional rendering  
-->  
  
<head>  
  
    <title>Patient Dashboard</title>  
  
    <!-- Page title for browser tab -->  
  
  
<link rel='stylesheet' href='/css/style.css' />  
    <!-- External stylesheet for consistent styling -->  
  
  
<style>  
  
    /*  
        Page-specific styles for dashboard layout  
        These styles create a modern, card-based dashboard design  
    */
```

```
/* Body styling */

body {
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    /* Modern font stack for consistent typography */

    background-color: #f4f6f8;
    /* Light gray background for subtle contrast */

    margin: 0;
    /* Remove default browser margins */

    padding: 20px;
    /* Add padding around page content */
}

/* Main dashboard container */

.dashboard-container {
    max-width: 1200px;
    /* Limit maximum width for better readability on large screens */

    margin: 0 auto;
    /* Center the container horizontally */

    background: white;
    /* White background for the main content area */

    padding: 30px;
    /* Padding inside the container */

    border-radius: 8px;
    /* Rounded corners for modern look */

    box-shadow: 0 0 10px rgba(0,0,0,0.1);
    /* Subtle shadow for depth */
}
```

```
/* Welcome section styling */

.welcome-section {
    text-align: center;
    /* Center text horizontally */
    margin-bottom: 30px;
    /* Space below welcome section */
    padding: 20px;
    /* Padding inside welcome section */
    background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
    /*
        CSS gradient background:
        - 135deg - Gradient angle (diagonal)
        - #667eea 0% - Start color (blue) at 0%
        - #764ba2 100% - End color (purple) at 100%
    */
    color: white;
    /* White text for contrast against gradient */
    border-radius: 8px;
    /* Rounded corners */
}

/* Quick actions grid layout */

.quick-actions {
    display: grid;
    /* CSS Grid for responsive layout */
    grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
    /*

```

Responsive grid columns:

- auto-fit - Automatically fit columns
- minmax(250px, 1fr) - Minimum 250px, maximum 1 fraction
- Creates responsive columns that adapt to screen size

*/

gap: 20px;

/* Space between grid items */

margin-bottom: 30px;

/* Space below quick actions */

}

/* Action card styling */

.action-card {

background: #f8f9fa;

/* Light gray background */

padding: 20px;

/* Padding inside cards */

border-radius: 8px;

/* Rounded corners */

text-align: center;

/* Center text in cards */

border: 1px solid #e9ecef;

/* Light border */

transition: transform 0.2s;

/* Smooth transform animation */

}

/* Action card hover effect */

```
.action-card:hover {  
    transform: translateY(-2px);  
    /* Move card up 2px on hover */  
    box-shadow: 0 4px 8px rgba(0,0,0,0.1);  
    /* Add shadow on hover for depth */  
}  
  
/* Action card links */  
  
.action-card a{  
    text-decoration: none;  
    /* Remove underline from links */  
    color: #007bff;  
    /* Blue color for links */  
    font-weight: bold;  
    /* Bold text for emphasis */  
}  
  
/* Patient information section */  
  
.patient-info {  
    background: #e3f2fd;  
    /* Light blue background */  
    padding: 20px;  
    /* Padding inside info section */  
    border-radius: 8px;  
    /* Rounded corners */  
    margin-bottom: 20px;  
    /* Space below patient info */  
}  
}
```

```
/* Logout button styling */

.logout-btn {
    background-color: #dc3545;
    /* Red background for logout button */
    color: white;
    /* White text */
    padding: 10px 20px;
    /* Padding inside button */
    border: none;
    /* Remove default border */
    border-radius: 4px;
    /* Rounded corners */
    cursor: pointer;
    /* Pointer cursor on hover */
    text-decoration: none;
    /* Remove underline from link */
    display: inline-block;
    /* Display as inline block for proper spacing */
}

/* Logout button hover effect */

.logout-btn:hover {
    background-color: #c82333;
    /* Darker red on hover */
}

</style>

</head>
```

```
<body>

    <!-- Include navigation bar fragment -->
    <div th:insert='fragments/navbar :: fragment'></div>
    <!--
        th:insert - Thymeleaf directive to include navbar fragment
        Provides consistent navigation across all pages
    -->

    <div class="dashboard-container">

        <!-- Welcome section -->
        <div class="welcome-section">
            <h1>Welcome to Your Dashboard</h1>
            <!-- Static welcome heading -->
            <p>Manage your appointments and health information</p>
            <!-- Static description -->
        </div>

        <!-- Patient information display -->
        <!--
            th:if="${patient}" - Thymeleaf conditional rendering
            Only displays this section if patient object exists in model
        -->
        <div class="patient-info" th:if="${patient}">
            <h3>Your Information</h3>
            <!-- Static heading for patient info -->

            <!-- Patient name -->
```

```
<p><strong>Name:</strong> <span  
th:text="${patient.name}"></span></p>  
<!--  
    th:text="${patient.name}" - Thymeleaf expression to display patient  
    name  
  
    ${patient.name} - Accesses name property of patient object  
  
    <strong> - Bold formatting for label  
  
-->  
  
<!-- Patient email -->  
  
<p><strong>Email:</strong> <span  
th:text="${patient.email}"></span></p>  
<!--  
    th:text="${patient.email}" - Displays patient email  
  
    Same pattern as name field  
  
-->  
  
<!-- Patient phone -->  
  
<p><strong>Phone:</strong> <span  
th:text="${patient.phone}"></span></p>  
<!-- Displays patient phone number -->  
  
<!-- Patient date of birth -->  
  
<p><strong>Date of Birth:</strong> <span  
th:text="${patient.dob}"></span></p>  
<!--  
    th:text="${patient.dob}" - Displays date of birth  
  
    LocalDate will be formatted according to locale  
  
-->
```

```
</div>
```

```
<!-- Quick action cards -->
```

```
<div class="quick-actions">
```

```
  <!-- Book Appointment card -->
```

```
    <div class="action-card">
```

```
      <h3>Book Appointment</h3>
```

```
      <!-- Static heading for card -->
```

```
      <p>Schedule a new appointment with a doctor</p>
```

```
      <!-- Static description -->
```

```
      <a href="/appointments/book">Book Now</a>
```

```
      <!--
```

```
        href="/appointments/book" - Links to appointment booking page
```

```
        Static link (no Thymeleaf needed for simple navigation)
```

```
      -->
```

```
    </div>
```

```
<!-- My Appointments card -->
```

```
<div class="action-card">
```

```
  <h3>My Appointments</h3>
```

```
  <!-- Static heading -->
```

```
  <p>View and manage your appointments</p>
```

```
  <!-- Static description -->
```

```
  <a href="/appointments/my">View Appointments</a>
```

```
  <!--
```

```
    href="/appointments/my" - Links to patient's appointment list
```

```
    This will show all appointments for the logged-in patient
```

```
  -->
```

```
</div>

<!-- Find Doctors card --&gt;
&lt;div class="action-card"&gt;
    &lt;h3&gt;Find Doctors&lt;/h3&gt;
    &lt;!-- Static heading --&gt;
    &lt;p&gt;Browse available doctors and their specializations&lt;/p&gt;
    &lt;!-- Static description --&gt;
    &lt;a href="/doctors"&gt;View Doctors&lt;/a&gt;
    &lt;!--
        href="/doctors" - Links to doctors listing page
        Allows patient to browse available doctors
    --&gt;
    &lt;/div&gt;
&lt;/div&gt;

<!-- Logout button --&gt;
&lt;div style="text-align: center;"&gt;
    &lt;!-- Inline style to center the logout button --&gt;
    &lt;a href="/patients/logout" class="logout-btn"&gt;Logout&lt;/a&gt;
    &lt;!--
        href="/patients/logout" - Links to logout endpoint
        class="logout-btn" - Applies logout button styling
        This will invalidate the session and redirect to home
    --&gt;
    &lt;/div&gt;
&lt;/div&gt;
&lt;/body&gt;</pre>
```

```
</html>

6. doctors.html (Doctors List

<!DOCTYPE html>

<!-- HTML5 document type declaration -->

<html xmlns:th='http://www.thymeleaf.org'>

<!--
    xmlns:th='http://www.thymeleaf.org' - Thymeleaf namespace
    Enables Thymeleaf expressions and iteration
-->

<head>
    <title>Doctors</title>
    <!-- Page title for browser tab -->

    <link rel='stylesheet' href='/css/style.css' />
    <!-- External stylesheet for consistent styling -->

    <style>
        /*
            Page-specific styles for doctors listing
            Creates a responsive grid layout for doctor cards
        */
        /* Body styling */
        body {
            font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
            /* Modern font stack */
            background-color: #f4f6f8;
            /* Light gray background */
        }
    </style>

```

```
margin: 0;  
/* Remove default margins */  
  
padding: 20px;  
/* Add padding around content */  
  
}  
  
  
/* Main doctors container */  
  
.doctors-container {  
  
max-width: 1200px;  
  
/* Limit maximum width for better readability */  
  
margin: 0 auto;  
/* Center the container horizontally */  
  
}  
  
  
/* Main heading styling */  
  
h1 {  
  
text-align: center;  
  
/* Center the heading */  
  
color: #2c3e50;  
  
/* Dark blue color */  
  
margin-bottom: 30px;  
/* Space below heading */  
  
}  
  
  
/* Doctors grid layout */  
  
.doctors-grid {  
  
display: grid;  
  
/* CSS Grid for responsive layout */
```

```
grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));  
/*  
  Responsive grid columns:  
    - auto-fit - Automatically fit columns  
    - minmax(300px, 1fr) - Minimum 300px, maximum 1 fraction  
    - Creates responsive columns that adapt to screen size  
*/  
  
gap: 20px;  
/* Space between grid items */  
}  
  
/* Individual doctor card styling */  
.doctor-card {  
background: white;  
/* White background for cards */  
padding: 20px;  
/* Padding inside cards */  
border-radius: 8px;  
/* Rounded corners */  
box-shadow: 0 2px 4px rgba(0,0,0,0.1);  
/* Subtle shadow for depth */  
border: 1px solid #e9ecef;  
/* Light border */  
}  
  
/* Doctor name heading */  
.doctor-card h3 {  
color: #2c3e50;
```

```
/* Dark blue color for doctor names */  
margin-bottom: 10px;  
/* Space below name */  
}
```

```
/* Specialization badge styling */
```

```
.specialization {  
background: #e3f2fd;  
/* Light blue background */  
color: #1976d2;  
/* Blue text color */  
padding: 5px 10px;  
/* Padding inside badge */  
border-radius: 15px;  
/* Rounded badge shape */  
font-size: 12px;  
/* Small font size */  
display: inline-block;  
/* Display as inline block */  
margin-bottom: 10px;  
/* Space below badge */  
}
```

```
/* Contact information styling */
```

```
.contact-info {  
margin: 10px 0;  
/* Margin around contact info */  
}
```

```
/* Contact info paragraphs */

.contact-info p {
    margin: 5px 0;
    /* Small margin between contact items */
    color: #666;
    /* Gray color for contact info */
}

/* View availability button */

.view-availability {
    background-color: #28a745;
    /* Green background */
    color: white;
    /* White text */
    padding: 8px 16px;
    /* Padding inside button */
    border: none;
    /* Remove default border */
    border-radius: 4px;
    /* Rounded corners */
    cursor: pointer;
    /* Pointer cursor on hover */
    text-decoration: none;
    /* Remove underline from link */
    display: inline-block;
    /* Display as inline block */
    margin-top: 10px;
}
```

```

/* Space above button */

}

/* View availability button hover effect */

.view-availability:hover{
    background-color: #218838;
    /* Darker green on hover */

}

</style>

</head>

<body>

<!-- Include navigation bar fragment -->

<div th:insert='fragments/navbar :: fragment'></div>

<!--

th:insert - Thymeleaf directive to include navbar fragment

Provides consistent navigation across all pages

-->

<div class="doctors-container">

<h1>Available Doctors</h1>

<!-- Static heading for doctors page -->

<!-- Doctors grid -->

<!--

th:if="${doctors != null and !doctors.empty}" - Thymeleaf conditional

Only displays grid if doctors list exists and is not empty

-->

```

```
<div class="doctors-grid" th:if="${doctors != null and !doctors.empty}">

<!--

    th:each="doctor : ${doctors}" - Thymeleaf iteration
    Iterates over each doctor in the doctors list
    Creates a doctor card for each doctor

-->

<div class="doctor-card" th:each="doctor : ${doctors}">

    <!-- Doctor name -->
    <h3 th:text="${doctor.name}">Doctor Name</h3>
    <!--
        th:text="${doctor.name}" - Displays doctor's name
        "Doctor Name" is fallback text (not shown when Thymeleaf processes)

-->

    <!-- Specialization badge -->
    <div class="specialization"
        th:text="${doctor.specialization}">Specialization</div>
    <!--
        th:text="${doctor.specialization}" - Displays doctor's specialization
        Styled as a badge for visual emphasis

-->

    <!-- Contact information -->
    <div class="contact-info">
        <!-- Email -->
        <p><strong>Email:</strong> <span
            th:text="${doctor.email}">email@example.com</span></p>
        <!--
            th:text="${doctor.email}" - Displays doctor's email
```

```

<strong> - Bold formatting for label
-->

<!-- Phone --&gt;

&lt;p&gt;&lt;strong&gt;Phone:&lt;/strong&gt; &lt;span
th:text="${doctor.phone}"&gt;+1234567890&lt;/span&gt;&lt;/p&gt;

&lt;!--
    th:text="${doctor.phone}" - Displays doctor's phone
    Same pattern as email
--&gt;

&lt;/div&gt;

<!-- View availability button --&gt;

&lt;!--
    th:href="@{/doctors/availability/{id}(id=${doctor.doctorId})}" -
Thymeleaf URL expression
    @{} - URL expression that generates proper URLs
    {id} - Path variable placeholder
    (id=${doctor.doctorId}) - Binds doctor.doctorId to {id} parameter
--&gt;

&lt;a th:href="@{/doctors/availability/{id}(id=${doctor.doctorId})}"
    class="view-availability"&gt;View Availability&lt;/a&gt;

&lt;!--
    class="view-availability" - Applies button styling
    "View Availability" - Button text
--&gt;

&lt;/div&gt;
&lt;/div&gt;
</pre>

```

```

<!-- No doctors message -->

<!--

    th:if="${doctors == null or doctors.empty}" - Thymeleaf conditional
    Only displays if doctors list is null or empty

-->

<div th:if="${doctors == null or doctors.empty}" style="text-align: center;
padding: 40px;">

    <!-- Inline style for centering and padding -->

    <h3>No doctors available at the moment.</h3>

    <!-- Static message when no doctors are available -->

    <p>Please check back later or contact the administrator.</p>

    <!-- Additional information for users -->

</div>

</div>

</body>

</html>

```

7. doctor-availability.html (Doctor Availability)

```

<!DOCTYPE html>

<!-- HTML5 document type declaration -->

<html xmlns:th='http://www.thymeleaf.org'>

<!--

    xmlns:th='http://www.thymeleaf.org' - Thymeleaf namespace
    Enables Thymeleaf expressions and conditional rendering

-->

<head>

    <title>Doctor Availability</title>

    <!-- Page title for browser tab -->

```

```
<link rel='stylesheet' href='/css/style.css' />

<!-- External stylesheet for consistent styling -->

<style>

/*
  Page-specific styles for doctor availability display
  Creates a responsive layout for time slots
*/

/* Body styling */

body {
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    /* Modern font stack for consistent typography */
    background-color: #f4f6f8;
    /* Light gray background */
    margin: 0;
    /* Remove default browser margins */
    padding: 20px;
    /* Add padding around content */
}

/* Main availability container */

.availability-container {
    max-width: 1000px;
    /* Limit maximum width for better readability */
    margin: 0 auto;
    /* Center the container horizontally */
}
```

```
/* Doctor information section */

.doctor-info {
    background: white;
    /* White background for doctor info */
    padding: 20px;
    /* Padding inside section */
    border-radius: 8px;
    /* Rounded corners */
    box-shadow: 0 2px 4px rgba(0,0,0,0.1);
    /* Subtle shadow for depth */
    margin-bottom: 20px;
    /* Space below doctor info */
}

/* Doctor name heading */

.doctor-info h2 {
    color: #2c3e50;
    /* Dark blue color for doctor name */
    margin-bottom: 10px;
    /* Space below name */
}

/* Specialization badge */

.specialization {
    background: #e3f2fd;
    /* Light blue background */
    color: #1976d2;
```

```
/* Blue text color */  
padding: 5px 10px;  
/* Padding inside badge */  
border-radius: 15px;  
/* Rounded badge shape */  
font-size: 12px;  
/* Small font size */  
display: inline-block;  
/* Display as inline block */  
margin-bottom: 10px;  
/* Space below badge */  
}  
  
/* Time slots section */
```

```
.time-slots {  
background: white;  
/* White background for time slots */  
padding: 20px;  
/* Padding inside section */  
border-radius: 8px;  
/* Rounded corners */  
box-shadow: 0 2px 4px rgba(0,0,0,0.1);  
/* Subtle shadow for depth */  
}
```

```
/* Time slots grid layout */  
.slots-grid {  
display: grid;
```

```
/* CSS Grid for responsive layout */
grid-template-columns: repeat(auto-fit, minmax(120px, 1fr));

/*
Responsive grid columns:
  - auto-fit - Automatically fit columns
  - minmax(120px, 1fr) - Minimum 120px, maximum 1 fraction
  - Creates responsive columns for time slots
*/

gap: 10px;

/* Space between grid items */
margin-top: 20px;

/* Space above slots grid */

}

/* Individual time slot styling */
.time-slot {
  padding: 10px;
  /* Padding inside time slots */
  border: 1px solid #ddd;
  /* Light border */
  border-radius: 4px;
  /* Rounded corners */
  text-align: center;
  /* Center text in slots */
  cursor: pointer;
  /* Pointer cursor for clickable slots */
  transition: all 0.2s;
  /* Smooth transition for hover effects */
```

```
}
```

```
/* Available time slot styling */
.time-slot.available {
    background-color: #d4edda;
    /* Light green background for available slots */
    border-color: #28a745;
    /* Green border */
    color: #155724;
    /* Dark green text */
}
```

```
/* Available time slot hover effect */
.time-slot.available:hover {
    background-color: #c3e6cb;
    /* Darker green on hover */
}
```

```
/* Booked time slot styling */
.time-slot.booked {
    background-color: #f8d7da;
    /* Light red background for booked slots */
    border-color: #dc3545;
    /* Red border */
    color: #721c24;
    /* Dark red text */
    cursor: not-allowed;
    /* Not-allowed cursor for booked slots */
}
```

```
}
```

```
/* Back button styling */

.back-btn {

    background-color: #6c757d;

    /* Gray background */

    color: white;

    /* White text */

    padding: 8px 16px;

    /* Padding inside button */

    border: none;

    /* Remove default border */

    border-radius: 4px;

    /* Rounded corners */

    cursor: pointer;

    /* Pointer cursor on hover */

    text-decoration: none;

    /* Remove underline from link */

    display: inline-block;

    /* Display as inline block */

    margin-bottom: 20px;

    /* Space below button */

}
```

```
/* Back button hover effect */

.back-btn:hover {

    background-color: #5a6268;

    /* Darker gray on hover */
```

```
        }

    </style>

</head>

<body>

    <!-- Include navigation bar fragment -->

    <div th:insert='fragments/navbar :: fragment'></div>

    <!--

        th:insert - Thymeleaf directive to include navbar fragment

        Provides consistent navigation across all pages

    -->

    <div class="availability-container">

        <!-- Back button -->

        <a href="/doctors" class="back-btn">← Back to Doctors</a>

        <!--

            href="/doctors" - Links back to doctors listing page

            class="back-btn" - Applies back button styling

            ← - Unicode arrow character for visual indication

        -->

        <!-- Doctor information -->

        <!--

            th:if="${doctor}" - Thymeleaf conditional rendering

            Only displays if doctor object exists in model

        -->

        <div class="doctor-info" th:if="${doctor}">

            <!-- Doctor name -->
```

```
<h2 th:text="${doctor.name}">Doctor Name</h2>
<!--
    th:text="${doctor.name}" - Displays doctor's name
    "Doctor Name" is fallback text (not shown when Thymeleaf processes)
-->

<!-- Specialization badge -->
<div class="specialization"
th:text="${doctor.specialization}">Specialization</div>
<!--
    th:text="${doctor.specialization}" - Displays doctor's specialization
    Styled as a badge for visual emphasis
-->

<!-- Doctor email -->
<p><strong>Email:</strong> <span
th:text="${doctor.email}">email@example.com</span></p>
<!--
    th:text="${doctor.email}" - Displays doctor's email
    <strong> - Bold formatting for label
-->

<!-- Doctor phone -->
<p><strong>Phone:</strong> <span
th:text="${doctor.phone}">+1234567890</span></p>
<!--
    th:text="${doctor.phone}" - Displays doctor's phone
    Same pattern as email
-->
```

```

</div>

<!-- Time slots section --&gt;
&lt;div class="time-slots"&gt;

    &lt;h3&gt;Available Time Slots&lt;/h3&gt;

    &lt;!-- Static heading for time slots --&gt;

    &lt;!-- Slots grid --&gt;
    &lt;!--

        th:if="${slots != null and !slots.empty}" - Thymeleaf conditional
        Only displays grid if slots list exists and is not empty
    --&gt;

    &lt;div class="slots-grid" th:if="${slots != null and !slots.empty}"&gt;
        &lt;!--

            th:each="slot : ${slots}" - Thymeleaf iteration
            Iterates over each time slot in the slots list
            Creates a time slot element for each slot
        --&gt;

        &lt;div th:each="slot : ${slots}"&gt;
            th:class="${slot.booked} ? 'time-slot booked' : 'time-slot available'"
            th:text="${slot.time}"&gt;
                09:00
        &lt;/div&gt;
        &lt;!--

            th:class="${slot.booked} ? 'time-slot booked' : 'time-slot available'" -
            Conditional CSS class
            If slot.booked is true, applies 'time-slot booked' class
            If slot.booked is false, applies 'time-slot available' class
        --&gt;
    &lt;/div&gt;
&lt;/div&gt;
</pre>

```

```
    th:text="${slot.time}" - Displays the time slot
    "09:00" is fallback text (not shown when Thymeleaf processes)

-->

</div>

<!-- No slots message --&gt;
&lt;!--
    th:if="${slots == null or slots.empty}" - Thymeleaf conditional
    Only displays if slots list is null or empty
--&gt;

&lt;div th:if="${slots == null or slots.empty}" style="text-align: center;
padding: 20px;"&gt;
    &lt;!-- Inline style for centering and padding --&gt;
    &lt;p&gt;No time slots available for this doctor.&lt;/p&gt;
    &lt;!-- Static message when no slots are available --&gt;
&lt;/div&gt;
&lt;/div&gt;

<!-- Booking instructions --&gt;
&lt;div style="background: #fff3cd; padding: 15px; border-radius: 4px; margin-
top: 20px;"&gt;
    &lt;!--
        Inline styles for instruction box:
        - background: #fff3cd - Light yellow background
        - padding: 15px - Padding inside box
        - border-radius: 4px - Rounded corners
        - margin-top: 20px - Space above box
    --&gt;
    &lt;h4&gt;How to Book:&lt;/h4&gt;</pre>
```

```
<!-- Static heading for instructions -->  
<p>1. Click on an available time slot (green)</p>  
<!-- Step 1 instruction -->  
<p>2. You will be redirected to the booking page</p>  
<!-- Step 2 instruction -->  
<p>3. Confirm your appointment details</p>  
<!-- Step 3 instruction -->  
<p><strong>Note:</strong> You must be logged in to book an  
appointment.</p>
```

```
<!--
```

```
    Important note about login requirement
```

```
    <strong> - Bold formatting for emphasis
```

```
    -->
```

```
  </div>
```

```
  </div>
```

```
</body>
```

```
</html>
```

8. appointment-book.html (Appointment Booking)

```
<!DOCTYPE html>
```

```
<!-- HTML5 document type declaration -->
```

```
<html xmlns:th='http://www.thymeleaf.org'>
```

```
<!--
```

```
  xmlns:th='http://www.thymeleaf.org' - Thymeleaf namespace
```

```
  Enables Thymeleaf expressions and URL generation
```

```
  -->
```

```
<head>
```

```
  <title>Book Appointment</title>
```

```
  <!-- Page title for browser tab -->
```

```
<link rel='stylesheet' href='/css/style.css' />  
<!-- External stylesheet for consistent styling -->  
  
<style>  
/*  
    Page-specific styles for appointment booking form  
    Creates a clean, user-friendly booking interface  
*/  
  
/* Body styling */  
  
body {  
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;  
    /* Modern font stack for consistent typography */  
    background-color: #f4f6f8;  
    /* Light gray background */  
    margin: 0;  
    /* Remove default browser margins */  
    padding: 20px;  
    /* Add padding around content */  
}  
  
/* Main booking container */  
.booking-container {  
    max-width: 800px;  
    /* Limit maximum width for better readability */  
    margin: 0 auto;  
    /* Center the container horizontally */
```

```
background: white;  
/* White background for form */  
  
padding: 30px;  
/* Padding inside container */  
  
border-radius: 8px;  
/* Rounded corners */  
  
box-shadow: 0 2px 4px rgba(0,0,0,0.1);  
/* Subtle shadow for depth */  
  
}
```

```
/* Main heading styling */  
  
h1 {  
  
text-align: center;  
/* Center the heading */  
  
color: #2c3e50;  
/* Dark blue color */  
  
margin-bottom: 30px;  
/* Space below heading */  
  
}
```

```
/* Form group styling */  
  
.form-group {  
  
margin-bottom: 20px;  
/* Space below each form group */  
  
}
```

```
/* Label styling */  
  
label {
```

```
display: block;  
/* Display as block element */  
  
margin-bottom: 5px;  
/* Space below label */  
  
font-weight: bold;  
/* Bold text for labels */  
  
color: #333;  
/* Dark gray color */  
  
}
```

```
/* Select and date input styling */  
  
select, input[type="date"] {  
  
width: 100%;  
/* Full width of container */  
  
padding: 10px;  
/* Padding inside inputs */  
  
border: 1px solid #ddd;  
/* Light border */  
  
border-radius: 4px;  
/* Rounded corners */  
  
font-size: 14px;  
/* Standard font size */  
  
}
```

```
/* Button styling */  
  
.btn {  
  
background-color: #007bff;  
/* Blue background */
```

```
color: white;  
/* White text */  
  
padding: 12px 24px;  
/* Padding inside button */  
  
border: none;  
/* Remove default border */  
  
border-radius: 4px;  
/* Rounded corners */  
  
cursor: pointer;  
/* Pointer cursor on hover */  
  
font-size: 16px;  
/* Standard font size */  
  
width: 100%;  
/* Full width of container */  
  
}
```

```
/* Button hover effect */  
  
.btn:hover {  
  
background-color: #0056b3;  
/* Darker blue on hover */  
  
}
```

```
/* Error message styling */  
  
.error-message {  
  
background-color: #f8d7da;  
/* Light red background */  
  
color: #721c24;  
/* Dark red text */
```

```
padding: 10px;  
/* Padding inside error box */  
  
border-radius: 4px;  
/* Rounded corners */  
  
margin-bottom: 20px;  
/* Space below error message */  
  
}  
  
  
/* Information box styling */  
  
.info-box {  
  
background-color: #d1ecf1;  
/* Light blue background */  
  
color: #0c5460;  
/* Dark blue text */  
  
padding: 15px;  
/* Padding inside info box */  
  
border-radius: 4px;  
/* Rounded corners */  
  
margin-bottom: 20px;  
/* Space below info box */  
  
}  
  
</style>  
  
</head>  
  
  
<body>  
<!-- Include navigation bar fragment --&gt;<br/>  
<div th:insert='fragments/navbar :: fragment'></div>  
  
!---
```

th:insert - Thymeleaf directive to include navbar fragment

Provides consistent navigation across all pages

-->

```
<div class="booking-container">

    <h1>Book an Appointment</h1>

    <!-- Static heading for booking page -->

    <!-- Error message -->
    <!--
        th:if="${param.error}" - Thymeleaf conditional for URL parameters
        Only displays if 'error' parameter exists in URL
        This shows when there's a booking error
    -->
    <div th:if="${param.error}" class="error-message">
        <strong>Error:</strong> There was a problem with your booking. Please try
        again.
        <!--
            <strong> - Bold formatting for "Error:" label
            Static error message for booking failures
        -->
        </div>

    <!-- Information box -->
    <div class="info-box">
        <h4>Booking Instructions:</h4>
        <!-- Static heading for instructions -->
        <p>1. Select a doctor from the list below</p>
```

```

<!-- Step 1 instruction -->
<p>2. Choose your preferred date</p>

<!-- Step 2 instruction -->
<p>3. Click "Check Availability" to see available time slots</p>

<!-- Step 3 instruction -->
<p>4. Select a time slot and confirm your appointment</p>

<!-- Step 4 instruction -->

</div>

<!-- Booking form -->
<!--
    th:action="@{/appointments/slots}" - Thymeleaf URL expression for form
    action
        @{} - URL expression that generates proper URLs
        method="get" - HTTP GET method for form submission
-->
<form th:action="@{/appointments/slots}" method="get">
    <!-- Doctor selection form group -->
    <div class="form-group">
        <label for="doctorId">Select Doctor:</label>
        <!--
            for="doctorId" - Associates label with select element
            Improves accessibility and user experience
-->
        <select id="doctorId" name="doctorId" required>
            <!--
                id="doctorId" - Unique identifier for select element
                name="doctorId" - Parameter name sent to server

```

```

required - HTML5 validation attribute

-->

<option value="">Choose a doctor...</option>

<!--

    Default option with empty value

    Provides placeholder text for user

-->

<!--

    th:each="doctor : ${doctors}" - Thymeleaf iteration

    Iterates over each doctor in the doctors list

    Creates an option for each doctor

-->

<option th:each="doctor : ${doctors}"

        th:value="${doctor.doctorId}"
        th:text="${doctor.name + ' - ' + doctor.specialization}">

    Doctor Name - Specialization

</option>

<!--

    th:value="${doctor.doctorId}" - Sets option value to doctor ID

    th:text="${doctor.name + ' - ' + doctor.specialization}" - Displays
doctor name and specialization

    "Doctor Name - Specialization" is fallback text (not shown when
Thymeleaf processes)

-->

</select>

</div>

<!-- Date selection form group -->

<div class="form-group">

```

```
<label for="date">Select Date:</label>

<!--

    for="date" - Associates label with date input

    Improves accessibility

-->

<input type="date" id="date" name="date" required

        th:min="#{temporals.format(temporals.createToday(), 'yyyy-MM-
dd')}">

<!--

    type="date" - HTML5 date picker

    id="date" - Unique identifier for input

    name="date" - Parameter name sent to server

    required - HTML5 validation attribute

    th:min="#{temporals.format(temporals.createToday(), 'yyyy-MM-
dd')}" - Thymeleaf expression for minimum date

    #temporals.createToday() - Creates today's date

    #temporals.format() - Formats date as yyyy-MM-dd

    This prevents users from selecting past dates

-->

</div>

<!-- Submit button -->

<button type="submit" class="btn">Check Availability</button>

<!--

    type="submit" - Submits the form when clicked

    class="btn" - Applies button styling

    "Check Availability" - Button text

-->

</form>
```

```
<!-- Back to dashboard link -->

<div style="text-align: center; margin-top: 20px;">
    <!-- Inline style for centering and spacing -->
    <a href="/patients/dashboard" style="color: #007bff; text-decoration: none;">← Back to Dashboard</a>
    <!--
        href="/patients/dashboard" - Links back to patient dashboard
        style="color: #007bff; text-decoration: none;" - Inline styles for link
        appearance
        ← - Unicode arrow character for visual indication
    -->
    </div>
</div>

<!-- JavaScript for date validation -->
<script>
    //
    // Client-side JavaScript for additional date validation
    // This runs in the browser to enhance user experience
    //

    // Set minimum date to today
    document.getElementById('date').min = new
Date().toISOString().split('T')[0];
    //
    // document.getElementById('date') - Gets the date input element
    // .min - Sets the minimum date attribute
    // new Date() - Current date and time

```

```
// toISOString() - Converts to ISO string format (YYYY-MM-
DDTHH:mm:ss.sssZ)

// split('T')[0] - Splits on 'T' and takes first part (date only)

// This prevents users from selecting past dates

//
```

```
</script>
```

```
</body>
```

```
</html>
```

9. appointment-slots.html (Available Time Slots)

```
<!DOCTYPE html>
```

```
<!-- HTML5 document type declaration -->
```

```
<html xmlns:th='http://www.thymeleaf.org'>
```

```
<!--
```

```
xmlns:th='http://www.thymeleaf.org' - Thymeleaf namespace
```

```
Enables Thymeleaf expressions and URL generation
```

```
-->
```

```
<head>
```

```
<title>Available Time Slots</title>
```

```
<!-- Page title for browser tab -->
```

```
<link rel='stylesheet' href='/css/style.css' />
```

```
<!-- External stylesheet for consistent styling -->
```

```
<style>
```

```
/*
```

```
Page-specific styles for time slots display
```

```
Creates an interactive grid of time slots
```

```
*/
```

```
/* Body styling */

body {
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    /* Modern font stack for consistent typography */

    background-color: #f4f6f8;
    /* Light gray background */

    margin: 0;
    /* Remove default browser margins */

    padding: 20px;
    /* Add padding around content */

}

/* Main slots container */

.slots-container {
    max-width: 1000px;
    /* Limit maximum width for better readability */

    margin: 0 auto;
    /* Center the container horizontally */

}

/* Doctor information section */

.doctor-info {
    background: white;
    /* White background for doctor info */

    padding: 20px;
    /* Padding inside section */

    border-radius: 8px;
}
```

```
/* Rounded corners */
box-shadow: 0 2px 4px rgba(0,0,0,0.1);

/* Subtle shadow for depth */
margin-bottom: 20px;

/* Space below doctor info */

}

/* Time slots section */
.slots-section {
background: white;
/* White background for time slots */
padding: 20px;
/* Padding inside section */
border-radius: 8px;
/* Rounded corners */
box-shadow: 0 2px 4px rgba(0,0,0,0.1);
/* Subtle shadow for depth */
}

/* Time slots grid layout */
.slots-grid {
display: grid;
/* CSS Grid for responsive layout */
grid-template-columns: repeat(auto-fit, minmax(150px, 1fr));
/*
Responsive grid columns:
- auto-fit - Automatically fit columns
- minmax(150px, 1fr) - Minimum 150px, maximum 1 fraction
```

```
- Creates responsive columns for time slots

*/
gap: 15px;
/* Space between grid items */

margin-top: 20px;
/* Space above slots grid */

}

/* Individual time slot styling */

.time-slot {
padding: 15px;
/* Padding inside time slots */
border: 2px solid #ddd;
/* Thick border for better visibility */
border-radius: 8px;
/* Rounded corners */
text-align: center;
/* Center text in slots */
cursor: pointer;
/* Pointer cursor for clickable slots */
transition: all 0.2s;
/* Smooth transition for hover effects */
text-decoration: none;
/* Remove underline from links */
color: inherit;
/* Inherit text color from parent */

}
```

```
/* Available time slot styling */

.time-slot.available {

    background-color: #d4edda;

    /* Light green background for available slots */

    border-color: #28a745;

    /* Green border */

    color: #155724;

    /* Dark green text */

}

/* Available time slot hover effect */

.time-slot.available:hover {

    background-color: #c3e6cb;

    /* Darker green on hover */

    transform: translateY(-2px);

    /* Move slot up 2px on hover for depth effect */

}

/* Booked time slot styling */

.time-slot.booked {

    background-color: #f8d7da;

    /* Light red background for booked slots */

    border-color: #dc3545;

    /* Red border */

    color: #721c24;

    /* Dark red text */

    cursor: not-allowed;

    /* Not-allowed cursor for booked slots */
}
```

```
    opacity: 0.6;  
    /* Reduce opacity for disabled appearance */  
}  
  
/* Back button styling */  
.back-btn {  
    background-color: #6c757d;  
    /* Gray background */  
    color: white;  
    /* White text */  
    padding: 8px 16px;  
    /* Padding inside button */  
    border: none;  
    /* Remove default border */  
    border-radius: 4px;  
    /* Rounded corners */  
    cursor: pointer;  
    /* Pointer cursor on hover */  
    text-decoration: none;  
    /* Remove underline from link */  
    display: inline-block;  
    /* Display as inline block */  
    margin-bottom: 20px;  
    /* Space below button */  
}  
  
/* Back button hover effect */  
.back-btn:hover {
```

```
background-color: #5a6268;  
/* Darker gray on hover */  
}  
  
/* No slots message styling */  
.no-slots {  
text-align: center;  
/* Center text horizontally */  
padding: 40px;  
/* Padding inside message area */  
color: #6c757d;  
/* Gray color for message */  
}  
</style>  
</head>  
  
<body>  
<!-- Include navigation bar fragment --&gt;<br/><div th:insert='fragments/navbar :: fragment'></div>  
!--  
th:insert - Thymeleaf directive to include navbar fragment  
Provides consistent navigation across all pages  
-->  
  
<div class="slots-container">  
<!-- Back button --&gt;<br/><a href="/appointments/book" class="back-btn">← Back to Booking</a>  
!--
```

`href="/appointments/book" - Links back to appointment booking page`

`class="back-btn" - Applies back button styling`

`< - Unicode arrow character for visual indication`

`-->`

`<!-- Doctor information -->`

`<!--`

`th:if="${doctor}" - Thymeleaf conditional rendering`

`Only displays if doctor object exists in model`

`-->`

`<div class="doctor-info" th:if="${doctor}">`

`<!-- Doctor name -->`

`<h2 th:text="${doctor.name}">Doctor Name</h2>`

`<!--`

`th:text="${doctor.name}" - Displays doctor's name`

`"Doctor Name" is fallback text (not shown when Thymeleaf processes)`

`-->`

`<!-- Doctor specialization -->`

`<p>Specialization: <span`

`th:text="${doctor.specialization}">Specialization</p>`

`<!--`

`th:text="${doctor.specialization}" - Displays doctor's specialization`

` - Bold formatting for label`

`-->`

`<!-- Appointment date -->`

`<p>Date: 2025-01-15</p>`

```

<!--

    th:text="${slots[0].date}" - Displays date from first slot

    th:if="${slots != null and !slots.empty}" - Only displays if slots exist

    "2025-01-15" is fallback text (not shown when Thymeleaf processes)

-->

</div>

<!-- Time slots section -->

<div class="slots-section">

    <h3>Available Time Slots</h3>

    <!-- Static heading for time slots -->

    <!-- Slots grid -->

    <!--

        th:if="${slots != null and !slots.empty}" - Thymeleaf conditional

        Only displays grid if slots list exists and is not empty

    -->

    <div class="slots-grid" th:if="${slots != null and !slots.empty}">

        <!--

            th:each="slot : ${slots}" - Thymeleaf iteration

            Iterates over each time slot in the slots list

            Creates a time slot link for each slot

        -->

        <a th:each="slot : ${slots}">

            th:class="${slot.booked} ? 'time-slot booked' : 'time-slot available'"

            th:href="${slot.booked} ? '#' :

@{/appointments/save(doctorId=${doctor.doctorId}, date=${slot.date},
time=${slot.time})}"

            th:onclick="${slot.booked} ? 'return false;' : """

```

```

th:text="${slot.time}">
09:00
</a>
<!--
    th:class="${slot.booked} ? 'time-slot booked' : 'time-slot available'" -
Conditional CSS class

    If slot.booked is true, applies 'time-slot booked' class
    If slot.booked is false, applies 'time-slot available' class

th:href="${slot.booked} ? '#' :
@{/appointments/save(doctorId=${doctor.doctorId}, date=${slot.date},
time=${slot.time})}" - Conditional href

    If slot.booked is true, href is '#' (no action)
    If slot.booked is false, href is the save appointment URL with
parameters

    @{} - Thymeleaf URL expression
    doctorId=${doctor.doctorId} - Passes doctor ID as parameter
    date=${slot.date} - Passes slot date as parameter
    time=${slot.time} - Passes slot time as parameter

th:onclick="${slot.booked} ? 'return false;' : "" - Conditional onclick

    If slot.booked is true, prevents default action
    If slot.booked is false, no onclick handler

th:text="${slot.time}" - Displays the time slot
"09:00" is fallback text (not shown when Thymeleaf processes)

-->
</div>

```

```
<!-- No slots message -->

<!--
    th:if="${slots == null or slots.empty}" - Thymeleaf conditional
    Only displays if slots list is null or empty
-->

<div th:if="${slots == null or slots.empty}" class="no-slots">
    <h4>No time slots available</h4>
    <!-- Static heading for no slots message -->
    <p>There are no available time slots for this doctor on the selected date.</p>
    <!-- Explanation message -->
    <p>Please try selecting a different date or doctor.</p>
    <!-- Suggestion for user action -->
</div>
</div>

<!-- Legend -->
<div style="background: #f8f9fa; padding: 15px; border-radius: 4px; margin-top: 20px;">
    <!--
        Inline styles for legend box:
        - background: #f8f9fa - Light gray background
        - padding: 15px - Padding inside box
        - border-radius: 4px - Rounded corners
        - margin-top: 20px - Space above box
    -->
    <h4>Legend:</h4>
    <!-- Static heading for legend -->
```

```
<p><span style="color: #28a745; font-weight: bold;">Green slots</span> -  
Available for booking</p>  
  
<!--  
  
Explanation for green slots  
  
style="color: #28a745; font-weight: bold;" - Inline styles for green, bold  
text  
  
-->  
  
<p><span style="color: #dc3545; font-weight: bold;">Red slots</span> -  
Already booked</p>  
  
<!--  
  
Explanation for red slots  
  
style="color: #dc3545; font-weight: bold;" - Inline styles for red, bold text  
  
-->  
  
</div>  
  
</div>  
  
</body>  
  
</html>
```

10. appointment-list.html (Patient's Appointments)

```
<!DOCTYPE html>  
  
<!-- HTML5 document type declaration -->  
  
<html xmlns:th='http://www.thymeleaf.org'>  
  
<!--  
  
xmlns:th='http://www.thymeleaf.org' - Thymeleaf namespace  
  
Enables Thymeleaf expressions and conditional rendering  
  
-->  
  
<head>  
  
<title>My Appointments</title>  
  
<!-- Page title for browser tab -->
```

```
<link rel='stylesheet' href='/css/style.css' />

<!-- External stylesheet for consistent styling -->

<style>

/*
  Page-specific styles for appointment list
  Creates a card-based layout for appointments
*/

/* Body styling */

body {
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    /* Modern font stack for consistent typography */
    background-color: #f4f6f8;
    /* Light gray background */
    margin: 0;
    /* Remove default browser margins */
    padding: 20px;
    /* Add padding around content */
}

/* Main appointments container */

.appointments-container{
    max-width: 1000px;
    /* Limit maximum width for better readability */
    margin: 0 auto;
    /* Center the container horizontally */
}
```

```
/* Main heading styling */

h1 {
    text-align: center;
    /* Center the heading */
    color: #2c3e50;
    /* Dark blue color */
    margin-bottom: 30px;
    /* Space below heading */
}

/* Individual appointment card styling */

.appointment-card {
    background: white;
    /* White background for cards */
    padding: 20px;
    /* Padding inside cards */
    border-radius: 8px;
    /* Rounded corners */
    box-shadow: 0 2px 4px rgba(0,0,0,0.1);
    /* Subtle shadow for depth */
    margin-bottom: 20px;
    /* Space below each card */
    border-left: 4px solid #007bff;
    /* Blue left border for default state */
}

/* Booked appointment card styling */
```

```
.appointment-card.booked {  
    border-left-color: #28a745;  
    /* Green left border for booked appointments */  
}  
  
/* Canceled appointment card styling */  
.appointment-card.canceled {  
    border-left-color: #dc3545;  
    /* Red left border for canceled appointments */  
}  
  
/* Completed appointment card styling */  
.appointment-card.completed {  
    border-left-color: #6c757d;  
    /* Gray left border for completed appointments */  
}  
  
/* Appointment header styling */  
.appointment-header {  
    display: flex;  
    /* Flexbox layout for header */  
    justify-content: space-between;  
    /* Space between appointment ID and status */  
    align-items: center;  
    /* Center items vertically */  
    margin-bottom: 15px;  
    /* Space below header */  
}
```

```
/* Appointment ID styling */

.appointment-id {
    font-weight: bold;
    /* Bold text for appointment ID */
    color: #6c757d;
    /* Gray color for ID */
}
```

```
/* Status badge styling */

.status-badge {
    padding: 4px 8px;
    /* Padding inside badge */
    border-radius: 12px;
    /* Rounded badge shape */
    font-size: 12px;
    /* Small font size */
    font-weight: bold;
    /* Bold text */
    text-transform: uppercase;
    /* Uppercase text */
}
```

```
/* Booked status badge styling */

.status-badge.booked {
    background-color: #d4edda;
    /* Light green background */
    color: #155724;
```

```
/* Dark green text */

}

/* Canceled status badge styling */
.status-badge.canceled {
    background-color: #f8d7da;
    /* Light red background */
    color: #721c24;
    /* Dark red text */
}

/* Completed status badge styling */
.status-badge.completed {
    background-color: #d1ecf1;
    /* Light blue background */
    color: #0c5460;
    /* Dark blue text */
}

/* Appointment details grid layout */
.appointment-details {
    display: grid;
    /* CSS Grid for responsive layout */
    grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
    /*

        Responsive grid columns:
        - auto-fit - Automatically fit columns
        - minmax(200px, 1fr) - Minimum 200px, maximum 1 fraction
    */
}
```

```
- Creates responsive columns for appointment details

*/
gap: 15px;
/* Space between grid items */

}

/* Individual detail item styling */

.detail-item {
margin-bottom: 10px;
/* Space below each detail item */

}

/* Detail label styling */

.detail-label {
font-weight: bold;
/* Bold text for labels */
color: #333;
/* Dark gray color */

}

/* Detail value styling */

.detail-value {
color: #666;
/* Gray color for values */

}

/* Cancel button styling */

.cancel-btn {
```

```
background-color: #dc3545;  
/* Red background */  
  
color: white;  
/* White text */  
  
padding: 6px 12px;  
/* Padding inside button */  
  
border: none;  
/* Remove default border */  
  
border-radius: 4px;  
/* Rounded corners */  
  
cursor: pointer;  
/* Pointer cursor on hover */  
  
text-decoration: none;  
/* Remove underline from link */  
  
font-size: 12px;  
/* Small font size */  
  
}
```

```
/* Cancel button hover effect */  
  
.cancel-btn:hover {  
  
background-color: #c82333;  
/* Darker red on hover */  
  
}
```

```
/* No appointments message styling */  
  
.no-appointments {  
  
text-align: center;  
/* Center text horizontally */
```

```
padding: 40px;  
/* Padding inside message area */  
color: #6c757d;  
/* Gray color for message */  
}  
  
/* Back button styling */
```

```
.back-btn {  
background-color: #6c757d;  
/* Gray background */  
color: white;  
/* White text */  
padding: 8px 16px;  
/* Padding inside button */  
border: none;  
/* Remove default border */  
border-radius: 4px;  
/* Rounded corners */  
cursor: pointer;  
/* Pointer cursor on hover */  
text-decoration: none;  
/* Remove underline from link */  
display: inline-block;  
/* Display as inline block */  
margin-bottom: 20px;  
/* Space below button */  
}
```

```
/* Back button hover effect */

.back-btn:hover {
    background-color: #5a6268;
    /* Darker gray on hover */

}

</style>

</head>

<body>

    <!-- Include navigation bar fragment -->

    <div th:insert='fragments/navbar :: fragment'></div>

    <!--

        th:insert - Thymeleaf directive to include navbar fragment

        Provides consistent navigation across all pages

    -->

    <div class="appointments-container">

        <!-- Back button -->

        <a href="/patients/dashboard" class="back-btn">← Back to Dashboard</a>

        <!--

            href="/patients/dashboard" - Links back to patient dashboard

            class="back-btn" - Applies back button styling

            ← - Unicode arrow character for visual indication

        -->

        <h1>My Appointments</h1>

        <!-- Static heading for appointments page -->

    </div>

</body>
```

```
<!-- Appointments list -->

<!--

    th:if="${appointments != null and !appointments.empty}" - Thymeleaf
    conditional

        Only displays list if appointments exist and are not empty

-->

<div th:if="${appointments != null and !appointments.empty}">

    <!--

        th:each="appointment : ${appointments}" - Thymeleaf iteration
        Iterates over each appointment in the appointments list
        Creates an appointment card for each appointment

-->

<div th:each="appointment : ${appointments}"
    th:class=""appointment-card ' +
    ${appointment.status.name().toLowerCase()}>

    <!--

        th:class=""appointment-card ' +
        ${appointment.status.name().toLowerCase()} - Dynamic CSS class
        'appointment-card' - Base class for all appointment cards
        + ${appointment.status.name().toLowerCase()} - Adds status-specific
        class

        appointment.status.name() - Gets status enum name (BOOKED,
        CANCELED, COMPLETED)

        toLowerCase() - Converts to lowercase for CSS class

-->

    <!-- Appointment header -->
    <div class="appointment-header">
        <!-- Appointment ID -->
```

```

<span class="appointment-id">Appointment #<span
th:text="${appointment.appointmentId}">123</span></span>

<!--

    th:text="${appointment.appointmentId}" - Displays appointment ID
    "123" is fallback text (not shown when Thymeleaf processes)

-->

<!-- Status badge -->

<span th:class=""status-badge ' +
${appointment.status.name().toLowerCase()}"

    th:text="${appointment.status}">BOOKED</span>

<!--

    th:class=""status-badge ' +
${appointment.status.name().toLowerCase()} - Dynamic CSS class
    'status-badge' - Base class for status badges
    + ${appointment.status.name().toLowerCase()} - Adds status-
specific class

    th:text="${appointment.status}" - Displays appointment status
    "BOOKED" is fallback text (not shown when Thymeleaf processes)

-->

</div>

<!-- Appointment details -->

<div class="appointment-details">

    <!-- Doctor name -->

    <div class="detail-item">
        <div class="detail-label">Doctor:</div>
        <div class="detail-value" th:text="${appointment.doctor.name}">Dr.
        Smith</div>

```

```
<!--  
    th:text="${appointment.doctor.name}" - Displays doctor's name  
    "Dr. Smith" is fallback text (not shown when Thymeleaf processes)  
-->  
</div>  
  
<!-- Doctor specialization -->  
<div class="detail-item">  
    <div class="detail-label">Specialization:</div>  
    <div class="detail-value"  
        th:text="${appointment.doctor.specialization}">Cardiology</div>  
    <!--  
        th:text="${appointment.doctor.specialization}" - Displays doctor's  
        specialization  
        "Cardiology" is fallback text (not shown when Thymeleaf  
        processes)  
    -->  
</div>  
  
<!-- Appointment date -->  
<div class="detail-item">  
    <div class="detail-label">Date:</div>  
    <div class="detail-value"  
        th:text="${appointment.appointmentDate}">2025-01-15</div>  
    <!--  
        th:text="${appointment.appointmentDate}" - Displays  
        appointment date  
        LocalDate will be formatted according to locale  
        "2025-01-15" is fallback text (not shown when Thymeleaf  
        processes)
```

```
-->

</div>

<!-- Appointment time -->

<div class="detail-item">

    <div class="detail-label">Time:</div>

    <div class="detail-value"
th:text="${appointment.timeSlot}">09:00</div>

    <!--

        th:text="${appointment.timeSlot}" - Displays appointment time
        LocalTime will be formatted according to locale
        "09:00" is fallback text (not shown when Thymeleaf processes)

    -->

</div>

<!-- Appointment status -->

<div class="detail-item">

    <div class="detail-label">Status:</div>

    <div class="detail-value"
th:text="${appointment.status}">BOOKED</div>

    <!--

        th:text="${appointment.status}" - Displays appointment status
        "BOOKED" is fallback text (not shown when Thymeleaf processes)

    -->

</div>

</div>

<!-- Cancel button (only for booked appointments) -->

<!--
```

```
th:if="${appointment.status.name() == 'BOOKED'}" - Thymeleaf  
conditional
```

Only displays cancel button if appointment status is BOOKED

-->

```
<div th:if="${appointment.status.name() == 'BOOKED'}" style="margin-  
top: 15px;">
```

<!-- Inline style for spacing above cancel button -->

<a

```
th:href="@{/appointments/cancel/{id}(id=${appointment.appointmentId})}"
```

class="cancel-btn"

onclick="return confirm('Are you sure you want to cancel this
appointment?')">

Cancel Appointment

<!--

```
th:href="@{/appointments/cancel/{id}(id=${appointment.appointmentId})}" -  
Thymeleaf URL expression
```

@{} - URL expression that generates proper URLs

/appointments/cancel/{id} - URL pattern with path variable

(id=\${appointment.appointmentId}) - Passes appointment ID as path
variable

class="cancel-btn" - Applies cancel button styling

onclick="return confirm('Are you sure you want to cancel this
appointment?') - JavaScript confirmation

return confirm() - Shows confirmation dialog, returns true/false

If user clicks OK, form submits; if Cancel, form doesn't submit

-->

</div>

</div>

```

</div>

<!-- No appointments message --&gt;
&lt;!--
    th:if="${appointments == null or appointments.empty}" - Thymeleaf
conditional
    Only displays if appointments list is null or empty
--&gt;
&lt;div th:if="${appointments == null or appointments.empty}" class="no-
appointments"&gt;
    &lt;h3&gt;No appointments found&lt;/h3&gt;
    &lt;!-- Static heading for no appointments message --&gt;
    &lt;p&gt;You don't have any appointments yet.&lt;/p&gt;
    &lt;!-- Explanation message --&gt;
    &lt;p&gt;&lt;a href="/appointments/book" style="color: #007bff;"&gt;Book your first
appointment&lt;/a&gt;&lt;/p&gt;
    &lt;!--
        Link to book appointment page
        href="/appointments/book" - Links to appointment booking page
        style="color: #007bff;" - Inline style for blue link color
--&gt;
    &lt;/div&gt;
&lt;/div&gt;
&lt;/body&gt;
&lt;/html&gt;

11. admin-login.html (Admin Login)

&lt;!DOCTYPE html&gt;
&lt;!-- HTML5 document type declaration --&gt;
&lt;html xmlns:th='http://www.thymeleaf.org'&gt;
</pre>

```

```
<!--  
    xmlns:th='http://www.thymeleaf.org' - Thymeleaf namespace  
    Enables Thymeleaf expressions and conditional rendering  
-->  
  
<head>  
    <title>Admin Login</title>  
    <!-- Page title for browser tab -->  
  
    <link rel='stylesheet' href='/css/style.css' />  
    <!-- External stylesheet for consistent styling -->  
  
<style>  
    /*  
        Page-specific styles for admin login form  
        Creates a professional login interface for administrators  
    */  
  
    /* Body styling */  
    body {  
        font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;  
        /* Modern font stack for consistent typography */  
        background-color: #f4f6f8;  
        /* Light gray background */  
        margin: 0;  
        /* Remove default browser margins */  
        padding: 20px;  
        /* Add padding around content */  
    }  
}
```

```
/* Main login container */

.login-container {
    max-width: 400px;
    /* Limit maximum width for better readability */

    margin: 0 auto;
    /* Center the container horizontally */

    background: white;
    /* White background for form */

    padding: 30px;
    /* Padding inside container */

    border-radius: 8px;
    /* Rounded corners */

    box-shadow: 0 2px 4px rgba(0,0,0,0.1);
    /* Subtle shadow for depth */

    margin-top: 50px;
    /* Space above container */
}

/* Main heading styling */

h1 {
    text-align: center;
    /* Center the heading */

    color: #2c3e50;
    /* Dark blue color */

    margin-bottom: 30px;
    /* Space below heading */
}
```

```
/* Form group styling */

.form-group {
    margin-bottom: 20px;
    /* Space below each form group */
}

/* Label styling */

label {
    display: block;
    /* Display as block element */
    margin-bottom: 5px;
    /* Space below label */
    font-weight: bold;
    /* Bold text for labels */
    color: #333;
    /* Dark gray color */
}

/* Input field styling */

input[type="email"], input[type="password"] {
    width: 100%;
    /* Full width of container */
    padding: 10px;
    /* Padding inside inputs */
    border: 1px solid #ddd;
    /* Light border */
    border-radius: 4px;
}
```

```
/* Rounded corners */
font-size: 14px;

/* Standard font size */
box-sizing: border-box;

/* Include padding and border in width calculation */

}

/* Input focus styling */
input[type="email"]:focus, input[type="password"]:focus {
    outline: none;
    /* Remove default browser outline */
    border-color: #007bff;
    /* Blue border on focus */
    box-shadow: 0 0 0 2px rgba(0, 123, 255, 0.25);
    /* Blue glow effect on focus */
}

/* Button styling */
.btn {
    background-color: #007bff;
    /* Blue background */
    color: white;
    /* White text */
    padding: 12px 24px;
    /* Padding inside button */
    border: none;
    /* Remove default border */
    border-radius: 4px;
}
```

```
/* Rounded corners */
cursor: pointer;

/* Pointer cursor on hover */
font-size: 16px;

/* Standard font size */
width: 100%;

/* Full width of container */

}

/* Button hover effect */
.btn:hover {
background-color: #0056b3;

/* Darker blue on hover */

}

/* Error message styling */
.error-message {
background-color: #f8d7da;

/* Light red background */

color: #721c24;

/* Dark red text */

padding: 10px;

/* Padding inside error box */

border-radius: 4px;

/* Rounded corners */

margin-bottom: 20px;

/* Space below error message */

}
```

```
/* Back to home link styling */

.back-link {
    text-align: center;
    /* Center text horizontally */
    margin-top: 20px;
    /* Space above link */

}

/* Back link anchor styling */

.back-link a {
    color: #007bff;
    /* Blue color for link */
    text-decoration: none;
    /* Remove underline */

}

/* Back link hover effect */

.back-link a:hover {
    text-decoration: underline;
    /* Add underline on hover */

}

</style>

</head>

<body>

<!-- Include navigation bar fragment -->

<div th:insert='fragments/navbar :: fragment'></div>
```

```
<!--  
    th:insert - Thymeleaf directive to include navbar fragment  
    Provides consistent navigation across all pages  
-->  
  
<div class="login-container">  
    <h1>Admin Login</h1>  
    <!-- Static heading for admin login page -->  
  
    <!-- Error message -->  
    <!--  
        th:if="${param.error}" - Thymeleaf conditional for URL parameters  
        Only displays if 'error' parameter exists in URL  
        This shows when login fails  
    -->  
    <div th:if="${param.error}" class="error-message">  
        <strong>Error:</strong> Invalid email or password. Please try again.  
        <!--  
            <strong> - Bold formatting for "Error:" label  
            Static error message for login failures  
        -->  
    </div>  
  
    <!-- Login form -->  
    <!--  
        th:action="@{/admin/login}" - Thymeleaf URL expression for form action  
        @{} - URL expression that generates proper URLs  
        method="post" - HTTP POST method for form submission  
    -->
```

```
-->

<form th:action="@{/admin/login}" method="post">

    <!-- Email input form group -->

    <div class="form-group">

        <label for="email">Email:</label>

        <!--

            for="email" - Associates label with input element

            Improves accessibility and user experience

        -->

        <input type="email" id="email" name="email" required>

        <!--

            type="email" - HTML5 email input with validation

            id="email" - Unique identifier for input element

            name="email" - Parameter name sent to server

            required - HTML5 validation attribute

        -->

    </div>

    <!-- Password input form group -->

    <div class="form-group">

        <label for="password">Password:</label>

        <!--

            for="password" - Associates label with input element

            Improves accessibility

        -->

        <input type="password" id="password" name="password" required>

        <!--

            type="password" - Password input with hidden text

        -->

    </div>
```

```
    id="password" - Unique identifier for input element
    name="password" - Parameter name sent to server
    required - HTML5 validation attribute
    -->
</div>

<!-- Submit button -->
<button type="submit" class="btn">Login</button>
<!--
    type="submit" - Submits the form when clicked
    class="btn" - Applies button styling
    "Login" - Button text
    -->
</form>

<!-- Back to home link -->
<div class="back-link">
    <a href="/">← Back to Home</a>
    <!--
        href="/" - Links back to home page
        ← - Unicode arrow character for visual indication
        -->
    </div>
</div>
</body>
</html>

12. admin-dashboard.html (Admin Dashboard)
<!DOCTYPE html>
```

```
<!-- HTML5 document type declaration -->

<html xmlns:th='http://www.thymeleaf.org'>

<!--
    xmlns:th='http://www.thymeleaf.org' - Thymeleaf namespace
    Enables Thymeleaf expressions and conditional rendering
-->

<head>
    <title>Admin Dashboard</title>
    <!-- Page title for browser tab -->

    <link rel='stylesheet' href='/css/style.css' />
    <!-- External stylesheet for consistent styling -->

    <style>
        /*
            Page-specific styles for admin dashboard
            Creates a professional dashboard interface for administrators
        */
    /* Body styling */
    body {
        font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
        /* Modern font stack for consistent typography */
        background-color: #f4f6f8;
        /* Light gray background */
        margin: 0;
        /* Remove default browser margins */
        padding: 20px;
    }

```

```
/* Add padding around content */  
}  
  
/* Main dashboard container */  
.dashboard-container {  
    max-width: 1200px;  
    /* Limit maximum width for better readability */  
    margin: 0 auto;  
    /* Center the container horizontally */  
}  
  
/* Dashboard header styling */  
.dashboard-header {  
    background: white;  
    /* White background for header */  
    padding: 20px;  
    /* Padding inside header */  
    border-radius: 8px;  
    /* Rounded corners */  
    box-shadow: 0 2px 4px rgba(0,0,0,0.1);  
    /* Subtle shadow for depth */  
    margin-bottom: 30px;  
    /* Space below header */  
    display: flex;  
    /* Flexbox layout for header */  
    justify-content: space-between;  
    /* Space between welcome message and logout button */  
    align-items: center;
```

```
/* Center items vertically */

}

/* Welcome message styling */
.welcome-message h1 {
    color: #2c3e50;
    /* Dark blue color for heading */
    margin: 0;
    /* Remove default margin */
}

/* Welcome message paragraph styling */
.welcome-message p {
    color: #6c757d;
    /* Gray color for subtitle */
    margin: 5px 0 0 0;
    /* Space above paragraph */
}

/* Logout button styling */
.logout-btn {
    background-color: #dc3545;
    /* Red background */
    color: white;
    /* White text */
    padding: 8px 16px;
    /* Padding inside button */
    border: none;
}
```

```
/* Remove default border */
border-radius: 4px;

/* Rounded corners */
cursor: pointer;

/* Pointer cursor on hover */
text-decoration: none;

/* Remove underline from link */
font-size: 14px;

/* Standard font size */

}

/* Logout button hover effect */
.logout-btn:hover {
    background-color: #c82333;
    /* Darker red on hover */
}

/* Dashboard cards grid layout */
.dashboard-cards {
    display: grid;
    /* CSS Grid for responsive layout */
    grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
    /*
        Responsive grid columns:
        - auto-fit - Automatically fit columns
        - minmax(300px, 1fr) - Minimum 300px, maximum 1 fraction
        - Creates responsive columns for dashboard cards
    */
}
```

```
gap: 20px;  
/* Space between grid items */  
  
margin-bottom: 30px;  
/* Space below cards */  
  
}  
  
  
/* Individual dashboard card styling */  
  
.dashboard-card {  
  
background: white;  
/* White background for cards */  
  
padding: 20px;  
/* Padding inside cards */  
  
border-radius: 8px;  
/* Rounded corners */  
  
box-shadow: 0 2px 4px rgba(0,0,0,0.1);  
/* Subtle shadow for depth */  
  
text-align: center;  
/* Center text in cards */  
  
}  
  
  
/* Card icon styling */  
  
.card-icon {  
  
font-size: 48px;  
/* Large font size for icons */  
  
margin-bottom: 15px;  
/* Space below icon */  
  
}
```

```
/* Card heading styling */

.card-heading {
    color: #2c3e50;
    /* Dark blue color for headings */

    margin-bottom: 10px;
    /* Space below heading */

}

/* Card description styling */

.card-description {
    color: #6c757d;
    /* Gray color for descriptions */

    margin-bottom: 20px;
    /* Space below description */

}

/* Card button styling */

.card-btn {
    background-color: #007bff;
    /* Blue background */

    color: white;
    /* White text */

    padding: 10px 20px;
    /* Padding inside button */

    border: none;
    /* Remove default border */

    border-radius: 4px;
    /* Rounded corners */
```

```
cursor: pointer;  
/* Pointer cursor on hover */  
  
text-decoration: none;  
/* Remove underline from link */  
  
display: inline-block;  
/* Display as inline block */  
}  
  
  
/* Card button hover effect */  
  
.card-btn:hover {  
  
background-color: #0056b3;  
/* Darker blue on hover */  
}  
  
  
/* Quick stats section styling */  
  
.quick-stats {  
  
background: white;  
/* White background for stats section */  
  
padding: 20px;  
/* Padding inside section */  
  
border-radius: 8px;  
/* Rounded corners */  
  
box-shadow: 0 2px 4px rgba(0,0,0,0.1);  
/* Subtle shadow for depth */  
  
margin-bottom: 30px;  
/* Space below stats section */  
}
```

```
/* Stats grid layout */

.stats-grid {
    display: grid;
    /* CSS Grid for responsive layout */
    grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
    /*

        Responsive grid columns:
        - auto-fit - Automatically fit columns
        - minmax(200px, 1fr) - Minimum 200px, maximum 1 fraction
        - Creates responsive columns for stats
    */
    gap: 20px;
    /* Space between grid items */
}

/* Individual stat item styling */

.stat-item {
    text-align: center;
    /* Center text in stat items */
    padding: 15px;
    /* Padding inside stat items */
    background: #f8f9fa;
    /* Light gray background */
    border-radius: 4px;
    /* Rounded corners */
}

/* Stat number styling */
```

```
.stat-number {  
    font-size: 24px;  
    /* Large font size for numbers */  
    font-weight: bold;  
    /* Bold text for numbers */  
    color: #007bff;  
    /* Blue color for numbers */  
    margin-bottom: 5px;  
    /* Space below number */  
}  
  
/* Stat label styling */  
.stat-label {  
    color: #6c757d;  
    /* Gray color for labels */  
    font-size: 14px;  
    /* Small font size for labels */  
}  
</style>  
</head>  
  
<body>  
    <!-- Include navigation bar fragment -->  
    <div th:insert='fragments/navbar :: fragment'></div>  
    <!--  
        th:insert - Thymeleaf directive to include navbar fragment  
        Provides consistent navigation across all pages  
    -->
```

```
<div class="dashboard-container">

    <!-- Dashboard header -->

    <div class="dashboard-header">

        <!-- Welcome message -->

        <div class="welcome-message">

            <h1>Admin Dashboard</h1>

            <!-- Static heading for admin dashboard -->

            <p>Welcome, <span th:text="${admin.name}">Admin</span>!</p>

            <!--

                th:text="${admin.name}" - Displays admin's name

                "Admin" is fallback text (not shown when Thymeleaf processes)

            -->

        </div>

        <!-- Logout button -->

        <a href="/admin/logout" class="logout-btn">Logout</a>

        <!--

            href="/admin/logout" - Links to admin logout endpoint

            class="logout-btn" - Applies logout button styling

        -->

    </div>

    <!-- Quick stats section -->

    <div class="quick-stats">

        <h2>Quick Statistics</h2>

        <!-- Static heading for stats section -->
```

```
<!-- Stats grid -->

<div class="stats-grid">

    <!-- Total doctors stat -->

    <div class="stat-item">

        <div class="stat-number" th:text="${totalDoctors}">0</div>

        <!--
            th:text="${totalDoctors}" - Displays total number of doctors
            "0" is fallback text (not shown when Thymeleaf processes)
        -->

        <div class="stat-label">Total Doctors</div>

        <!-- Static label for doctors count -->

    </div>

    <!-- Total appointments stat -->

    <div class="stat-item">

        <div class="stat-number" th:text="${totalAppointments}">0</div>

        <!--
            th:text="${totalAppointments}" - Displays total number of
            appointments
            "0" is fallback text (not shown when Thymeleaf processes)
        -->

        <div class="stat-label">Total Appointments</div>

        <!-- Static label for appointments count -->

    </div>

    <!-- Today's appointments stat -->

    <div class="stat-item">

        <div class="stat-number" th:text="${todayAppointments}">0</div>
```

```
<!--  
    th:text="${todayAppointments}" - Displays today's appointments  
count  
    "0" is fallback text (not shown when Thymeleaf processes)  
-->  
  
<div class="stat-label">Today's Appointments</div>  
  
<!-- Static label for today's appointments count -->  
  
</div>  
  
  
<!-- Pending appointments stat -->  
  
<div class="stat-item">  
  
    <div class="stat-number" th:text="${pendingAppointments}">0</div>  
  
<!--  
    th:text="${pendingAppointments}" - Displays pending appointments  
count  
    "0" is fallback text (not shown when Thymeleaf processes)  
-->  
  
<div class="stat-label">Pending Appointments</div>  
  
<!-- Static label for pending appointments count -->  
  
</div>  
  
</div>  
  
  
<!-- Dashboard cards -->  
  
<div class="dashboard-cards">  
  
<!-- Manage doctors card -->  
  
<div class="dashboard-card">  
  
    <div class="card-icon">  </div>
```

```
<!-- Unicode doctor emoji for visual appeal -->
<h3 class="card-heading">Manage Doctors</h3>
<!-- Static heading for doctors management -->
<p class="card-description">Add, edit, or remove doctors from the system</p>
<!-- Description of doctors management functionality -->
<a href="/admin/addDoctor" class="card-btn">Add Doctor</a>
<!--
    href="/admin/addDoctor" - Links to add doctor page
    class="card-btn" - Applies card button styling
-->
</div>

<!-- View appointments card -->
<div class="dashboard-card">
    <div class="card-icon">CALENDAR</div>
    <!-- Unicode calendar emoji for visual appeal -->
    <h3 class="card-heading">View Appointments</h3>
    <!-- Static heading for appointments viewing -->
    <p class="card-description">View and manage all patient appointments</p>
    <!-- Description of appointments management functionality -->
    <a href="/admin/appointments" class="card-btn">View Appointments</a>
    <!--
        href="/admin/appointments" - Links to appointments list page
        class="card-btn" - Applies card button styling
    -->
</div>
```

```

<!-- System settings card -->

<div class="dashboard-card">

    <div class="card-icon">⚙️</div>

    <!-- Unicode gear emoji for visual appeal -->

    <h3 class="card-heading">System Settings</h3>

    <!-- Static heading for system settings -->

    <p class="card-description">Configure system preferences and
settings</p>

    <!-- Description of system settings functionality -->

    <a href="#" class="card-btn" onclick="alert('System settings coming
soon!')">Settings</a>

    <!--
        href="#" - Placeholder link (no actual page yet)
        class="card-btn" - Applies card button styling
        onclick="alert('System settings coming soon!')" - JavaScript alert for
placeholder
    -->

    </div>

</div>

</div>

</body>

</html>

13. admin-appointments.html (Admin Appointments Management)

<!DOCTYPE html>

<!-- HTML5 document type declaration -->

<html xmlns:th='http://www.thymeleaf.org'>

<!--
    xmlns:th='http://www.thymeleaf.org' - Thymeleaf namespace

```

Enables Thymeleaf expressions and conditional rendering

-->

<head>

<title>Admin - Appointments</title>

<!-- Page title for browser tab -->

<link rel='stylesheet' href='/css/style.css' />

<!-- External stylesheet for consistent styling -->

<style>

/*

Page-specific styles for admin appointments management

Creates a professional interface for managing all appointments

*/

/* Body styling */

body {

font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

/* Modern font stack for consistent typography */

background-color: #f4f6f8;

/* Light gray background */

margin: 0;

/* Remove default browser margins */

padding: 20px;

/* Add padding around content */

}

/* Main appointments container */

```
.appointments-container {  
    max-width: 1200px;  
    /* Limit maximum width for better readability */  
    margin: 0 auto;  
    /* Center the container horizontally */  
}  
  
/* Page header styling */  
  
.page-header {  
    background: white;  
    /* White background for header */  
    padding: 20px;  
    /* Padding inside header */  
    border-radius: 8px;  
    /* Rounded corners */  
    box-shadow: 0 2px 4px rgba(0,0,0,0.1);  
    /* Subtle shadow for depth */  
    margin-bottom: 30px;  
    /* Space below header */  
    display: flex;  
    /* Flexbox layout for header */  
    justify-content: space-between;  
    /* Space between heading and back button */  
    align-items: center;  
    /* Center items vertically */  
}  
  
/* Page heading styling */
```

```
.page-header h1 {  
    color: #2c3e50;  
    /* Dark blue color for heading */  
    margin: 0;  
    /* Remove default margin */  
}  
  
/* Back button styling */  
  
.back-btn {  
    background-color: #6c757d;  
    /* Gray background */  
    color: white;  
    /* White text */  
    padding: 8px 16px;  
    /* Padding inside button */  
    border: none;  
    /* Remove default border */  
    border-radius: 4px;  
    /* Rounded corners */  
    cursor: pointer;  
    /* Pointer cursor on hover */  
    text-decoration: none;  
    /* Remove underline from link */  
    display: inline-block;  
    /* Display as inline block */  
}  
  
/* Back button hover effect */
```

```
.back-btn:hover {  
    background-color: #5a6268;  
    /* Darker gray on hover */  
}  
  
/* Appointments table styling */  
.appointments-table {  
    background: white;  
    /* White background for table */  
    border-radius: 8px;  
    /* Rounded corners */  
    box-shadow: 0 2px 4px rgba(0,0,0,0.1);  
    /* Subtle shadow for depth */  
    overflow: hidden;  
    /* Hide table overflow */  
}  
  
/* Table styling */  
table {  
    width: 100%;  
    /* Full width of container */  
    border-collapse: collapse;  
    /* Collapse table borders */  
}  
  
/* Table header styling */  
th {  
    background-color: #f8f9fa;
```

```
/* Light gray background for headers */  
color: #495057;  
  
/* Dark gray text color */  
padding: 15px;  
  
/* Padding inside header cells */  
text-align: left;  
  
/* Left align text in headers */  
font-weight: bold;  
  
/* Bold text for headers */  
border-bottom: 2px solid #dee2e6;  
  
/* Bottom border for headers */  
}  
  
/* Table data cell styling */
```

```
td {  
padding: 15px;  
  
/* Padding inside data cells */  
border-bottom: 1px solid #dee2e6;  
  
/* Bottom border for data cells */  
vertical-align: top;  
  
/* Align content to top of cells */  
}
```

```
/* Table row hover effect */  
tr:hover {  
background-color: #f8f9fa;  
  
/* Light gray background on hover */  
}
```

```
/* Status badge styling */

.status-badge {

    padding: 4px 8px;

    /* Padding inside badge */

    border-radius: 12px;

    /* Rounded badge shape */

    font-size: 12px;

    /* Small font size */

    font-weight: bold;

    /* Bold text */

    text-transform: uppercase;

    /* Uppercase text */

}
```

```
/* Booked status badge styling */

.status-badge.booked {

    background-color: #d4edda;

    /* Light green background */

    color: #155724;

    /* Dark green text */

}
```

```
/* Canceled status badge styling */

.status-badge.canceled {

    background-color: #f8d7da;

    /* Light red background */

    color: #721c24;
```

```
/* Dark red text */  
}  
  
/* Completed status badge styling */  
.status-badge.completed {  
    background-color: #d1ecf1;  
    /* Light blue background */  
    color: #0c5460;  
    /* Dark blue text */  
}  
  
/* Action button styling */  
.action-btn {  
    background-color: #dc3545;  
    /* Red background */  
    color: white;  
    /* White text */  
    padding: 6px 12px;  
    /* Padding inside button */  
    border: none;  
    /* Remove default border */  
    border-radius: 4px;  
    /* Rounded corners */  
    cursor: pointer;  
    /* Pointer cursor on hover */  
    text-decoration: none;  
    /* Remove underline from link */  
    font-size: 12px;
```

```
/* Small font size */

}

/* Action button hover effect */

.action-btn:hover {
    background-color: #c82333;
    /* Darker red on hover */
}

/* No appointments message styling */

.no-appointments {
    text-align: center;
    /* Center text horizontally */
    padding: 40px;
    /* Padding inside message area */
    color: #6c757d;
    /* Gray color for message */
    background: white;
    /* White background for message */
    border-radius: 8px;
    /* Rounded corners */
    box-shadow: 0 2px 4px rgba(0,0,0,0.1);
    /* Subtle shadow for depth */
}

/* Patient info styling */

.patient-info {
    font-weight: bold;
}
```

```
/* Bold text for patient name */
color: #2c3e50;

/* Dark blue color */

}

/* Doctor info styling */
.doctor-info {
    color: #6c757d;
    /* Gray color for doctor info */
}

/* Date and time styling */
.datetime-info {
    color: #495057;
    /* Dark gray color for date/time */
}

</style>

</head>

<body>


<div th:insert='fragments/navbar :: fragment'></div>

<!--

th:insert - Thymeleaf directive to include navbar fragment

Provides consistent navigation across all pages

--&gt;

&lt;div class="appointments-container"&gt;</pre>
```

```

<!-- Page header -->

<div class="page-header">

    <h1>All Appointments</h1>

    <!-- Static heading for appointments page -->

    <a href="/admin/dashboard" class="back-btn">← Back to Dashboard</a>

    <!--

        href="/admin/dashboard" - Links back to admin dashboard

        class="back-btn" - Applies back button styling

        ← - Unicode arrow character for visual indication

    -->

</div>

<!-- Appointments table -->

<!--

    th:if="${appointments != null and !appointments.empty}" - Thymeleaf
    conditional

    Only displays table if appointments exist and are not empty

-->

<div th:if="${appointments != null and !appointments.empty}"
class="appointments-table">

    <table>

        <!-- Table header -->

        <thead>

            <tr>

                <th>Appointment ID</th>

                <!-- Static header for appointment ID -->

                <th>Patient</th>

                <!-- Static header for patient -->

                <th>Doctor</th>


```

```

<!-- Static header for doctor -->
<th>Date</th>
<!-- Static header for date -->
<th>Time</th>
<!-- Static header for time -->
<th>Status</th>
<!-- Static header for status -->
<th>Actions</th>
<!-- Static header for actions -->
</tr>
</thead>

<!-- Table body -->
<tbody>
<!--
    th:each="appointment : ${appointments}" - Thymeleaf iteration
    Iterates over each appointment in the appointments list
    Creates a table row for each appointment
-->
<tr th:each="appointment : ${appointments}">
    <!-- Appointment ID cell -->
    <td th:text="${appointment.appointmentId}">123</td>
    <!--
        th:text="${appointment.appointmentId}" - Displays appointment
        ID
        "123" is fallback text (not shown when Thymeleaf processes)
-->

```

```

<!-- Patient info cell -->

<td>

    <div class="patient-info"
th:text="${appointment.patient.name}">John Doe</div>

    <!--

        th:text="${appointment.patient.name}" - Displays patient's name

        "John Doe" is fallback text (not shown when Thymeleaf
processes)

    -->

    <div class="doctor-info"
th:text="${appointment.patient.email}">john@example.com</div>

    <!--

        th:text="${appointment.patient.email}" - Displays patient's email

        "john@example.com" is fallback text (not shown when Thymeleaf
processes)

    -->

    </td>

<!-- Doctor info cell -->

<td>

    <div class="patient-info"
th:text="${appointment.doctor.name}">Dr. Smith</div>

    <!--

        th:text="${appointment.doctor.name}" - Displays doctor's name

        "Dr. Smith" is fallback text (not shown when Thymeleaf
processes)

    -->

    <div class="doctor-info"
th:text="${appointment.doctor.specialization}">Cardiology</div>

    <!--

```

```
    th:text="${appointment.doctor.specialization}" - Displays  
doctor's specialization
```

```
    "Cardiology" is fallback text (not shown when Thymeleaf  
processes)
```

```
-->
```

```
</td>
```

```
<!-- Date cell -->
```

```
<td class="datetime-info"
```

```
th:text="${appointment.appointmentDate}">2025-01-15</td>
```

```
<!--
```

```
    th:text="${appointment.appointmentDate}" - Displays  
appointment date
```

```
LocalDate will be formatted according to locale
```

```
    "2025-01-15" is fallback text (not shown when Thymeleaf  
processes)
```

```
-->
```

```
<!-- Time cell -->
```

```
<td class="datetime-info"
```

```
th:text="${appointment.timeSlot}">09:00</td>
```

```
<!--
```

```
    th:text="${appointment.timeSlot}" - Displays appointment time
```

```
LocalTime will be formatted according to locale
```

```
    "09:00" is fallback text (not shown when Thymeleaf processes)
```

```
-->
```

```
<!-- Status cell -->
```

```
<td>
```

```

<span th:class=""status-badge ' +
${appointment.status.name().toLowerCase()}"

    th:text="${appointment.status}">BOOKED</span>

<!--

    th:class=""status-badge ' +
${appointment.status.name().toLowerCase()}> - Dynamic CSS class

    'status-badge ' - Base class for status badges

    + ${appointment.status.name().toLowerCase()} - Adds status-
specific class

    appointment.status.name() - Gets status enum name (BOOKED,
CANCELED, COMPLETED)

    toLowerCase() - Converts to lowercase for CSS class

    th:text="${appointment.status}" - Displays appointment status

    "BOOKED" is fallback text (not shown when Thymeleaf
processes)

-->

</td>

<!-- Actions cell -->

<td>

<!--

    th:if="${appointment.status.name() == 'BOOKED'}" - Thymeleaf
conditional

    Only displays cancel button if appointment status is BOOKED

-->

<a th:if="${appointment.status.name() == 'BOOKED'}"

th:href="@{/admin/cancel/{id}(id=${appointment.appointmentId})}"
class="action-btn"

```

```
    onclick="return confirm('Are you sure you want to cancel this
appointment?')">>

        Cancel

    </a>

    <!--

Thymeleaf URL expression

    @{} - URL expression that generates proper URLs

    /admin/cancel/{id} - URL pattern with path variable

    (id=${appointment.appointmentId}) - Passes appointment ID as
path variable

    class="action-btn" - Applies action button styling

    onclick="return confirm('Are you sure you want to cancel this
appointment?') - JavaScript confirmation

    return confirm() - Shows confirmation dialog, returns true/false

    If user clicks OK, form submits; if Cancel, form doesn't submit

-->

    <!--

Thymeleaf conditional

    Only displays message if appointment status is not BOOKED

-->

    <span th:if="${appointment.status.name() != 'BOOKED'}"
style="color: #6c757d; font-style: italic;">No actions
available</span>

    <!--

    Static message for appointments that cannot be canceled
```

italic text
-->
</td>
</tr>
</tbody>
</table>
</div>

<!-- No appointments message -->
<!--
 th:if="\${appointments == null or appointments.empty}" - Thymeleaf
conditional
Only displays if appointments list is null or empty
-->
<div th:if="\${appointments == null or appointments.empty}" class="no-
appointments">
 <h3>No appointments found</h3>
 <!-- Static heading for no appointments message -->
 <p>There are no appointments in the system yet.</p>
 <!-- Explanation message -->
 <p>Appointments will appear here once patients start booking them.</p>
 <!-- Additional information message -->
 </div>
</div>
</body>
</html>