The **Admin User Stories**:

**User Story 1**

**Title:**  
*As an admin, I want to log into the portal with my username and password, so that I can manage the platform securely.*

**Acceptance Criteria:**

1. The login form should accept a valid username and password.
2. Invalid credentials should display an error message without logging in the user.
3. Upon successful login, the admin should be redirected to the Admin Dashboard.

**Priority:** High  
**Story Points:** 3  
**Notes:**

* Consider implementing account lockout after multiple failed login attempts.
* Use Spring Security for authentication and session management.

**User Story 2**

**Title:**  
*As an admin, I want to log out of the portal, so that I can protect access to the system when I’m done.*

**Acceptance Criteria:**

1. A logout option should be clearly visible on the admin dashboard.
2. Clicking logout should invalidate the session and redirect the user to the login page.
3. The browser back button should not allow access to authenticated pages after logout.

**Priority:** High  
**Story Points:** 2  
**Notes:**

* Ensure CSRF protection is in place for the logout request.
* Session timeout should also be considered as an additional security measure.

**User Story 3**

**Title:**  
*As an admin, I want to add doctors to the portal, so that they can be assigned patients and manage appointments.*

**Acceptance Criteria:**

1. Admin should be able to access an “Add Doctor” form.
2. Submitting the form with valid data should create a new doctor profile.
3. The new doctor should appear in the doctor list immediately after creation.

**Priority:** High  
**Story Points:** 5  
**Notes:**

* Validate doctor details before submission (e.g., email format, medical license number).
* Assign default roles/permissions for newly added doctors.

**User Story 4**

**Title:**  
*As an admin, I want to delete a doctor’s profile from the portal, so that I can manage staff changes efficiently.*

**Acceptance Criteria:**

1. Admin should see a “Delete” option next to each doctor profile.
2. Clicking delete should prompt a confirmation modal to prevent accidental deletion.
3. Once confirmed, the doctor profile should be removed from the system.

**Priority:** Medium  
**Story Points:** 3  
**Notes:**

* Check for dependencies before deletion (e.g., scheduled appointments or prescription history).
* Consider soft deletion for audit/logging purposes.

**User Story 5**

**Title:**  
*As an admin, I want to run a stored procedure in the MySQL CLI to get the number of appointments per month, so that I can track usage statistics.*

**Acceptance Criteria:**

1. A stored procedure should exist in the MySQL database that aggregates appointment counts by month.
2. Admin should be able to execute the procedure via CLI and view the output.
3. The result should include year, month, and number of appointments.

**Priority:** Medium  
**Story Points:** 2  
**Notes:**

* Procedure might be named something like CALL get\_monthly\_appointment\_stats();
* Consider adding permissions or audit logging for stored procedure execution.

The **Patient User Stories**:

**User Story 1**

**Title:**  
*As a patient, I want to view a list of doctors without logging in, so that I can explore options before registering.*

**Acceptance Criteria:**

1. The home page or public view should list all available doctors.
2. Doctor information should include name, specialty, and availability.
3. No login should be required to access this list.

**Priority:** High  
**Story Points:** 3  
**Notes:**

* Consider using caching for faster access to public doctor data.
* A “Book Now” or “Sign Up to Book” button can guide users to registration.

**User Story 2**

**Title:**  
*As a patient, I want to sign up using my email and password, so that I can book appointments.*

**Acceptance Criteria:**

1. A registration form should collect patient details including email and password.
2. The system should validate the email format and enforce strong password rules.
3. Successful registration should redirect the user to the login or dashboard page.

**Priority:** High  
**Story Points:** 3  
**Notes:**

* Email confirmation step is optional but recommended.
* Duplicate emails should not be allowed.

**User Story 3**

**Title:**  
*As a patient, I want to log into the portal, so that I can manage my bookings.*

**Acceptance Criteria:**

1. Login form should authenticate with registered email and password.
2. Invalid credentials should display an error.
3. Upon successful login, the user should be redirected to their dashboard.

**Priority:** High  
**Story Points:** 2  
**Notes:**

* Use Spring Security for authentication.
* Store session data securely and consider "Remember Me" option.

**User Story 4**

**Title:**  
*As a patient, I want to log out of the portal, so that I can secure my account.*

**Acceptance Criteria:**

1. The logout button should be visible on all authenticated views.
2. Clicking it should terminate the session and redirect to the home or login page.
3. After logout, browser navigation should not expose private content.

**Priority:** High  
**Story Points:** 2  
**Notes:**

* Ensure logout endpoint uses POST for CSRF protection.
* Implement session expiration timeout as a fallback.

**User Story 5**

**Title:**  
*As a patient, I want to book an hour-long appointment with a doctor, so that I can consult with them.*

**Acceptance Criteria:**

1. Patients should see available time slots for each doctor.
2. Selecting a slot and confirming should create an appointment.
3. The system should prevent overlapping or double-booked slots.

**Priority:** High  
**Story Points:** 5  
**Notes:**

* Include validations for doctor availability.
* Appointment duration should default to one hour unless otherwise specified.

**User Story 6**

**Title:**  
*As a patient, I want to view my upcoming appointments, so that I can prepare accordingly.*

**Acceptance Criteria:**

1. Patients should have access to a list of their future appointments.
2. Each item should include date, time, doctor, and location or mode (e.g., telehealth).
3. Only future appointments should be displayed; past ones can go in a history tab.

**Priority:** Medium  
**Story Points:** 3  
**Notes:**

* Consider sorting appointments chronologically.
* Use pagination or infinite scroll if the list grows large.

The **Doctor User Stories**:

**User Story 1**

**Title:**  
*As a doctor, I want to log into the portal, so that I can manage my appointments.*

**Acceptance Criteria:**

1. The login form should allow authentication with the doctor’s credentials.
2. On successful login, the doctor should be redirected to their dashboard.
3. Failed logins should provide an appropriate error message.

**Priority:** High  
**Story Points:** 2  
**Notes:**

* Use role-based access control to ensure doctors only see their own data.
* Implement two-factor authentication optionally for added security.

**User Story 2**

**Title:**  
*As a doctor, I want to log out of the portal, so that I can protect my data.*

**Acceptance Criteria:**

1. A logout button should be clearly visible on the dashboard.
2. Clicking logout should terminate the session and redirect to the login or home page.
3. Post-logout, restricted content should not be accessible via the back button.

**Priority:** High  
**Story Points:** 2  
**Notes:**

* Logout should trigger a full session invalidation.
* Ensure CSRF protection is in place.

**User Story 3**

**Title:**  
*As a doctor, I want to view my appointment calendar, so that I can stay organized.*

**Acceptance Criteria:**

1. Doctors should see a calendar view with all booked appointments.
2. Each entry should show the patient’s name, time, and purpose of visit.
3. Appointments should be filterable by date and searchable by patient name.

**Priority:** High  
**Story Points:** 4  
**Notes:**

* Consider integrating with a JavaScript calendar like FullCalendar in the front end.
* Ensure time zones are handled correctly if applicable.

**User Story 4**

**Title:**  
*As a doctor, I want to mark my unavailability, so that patients only see available slots.*

**Acceptance Criteria:**

1. Doctors should have access to a form or calendar to define unavailable times.
2. Marked slots should be blocked from patient booking view.
3. Past unavailability settings should remain in the log for reference.

**Priority:** High  
**Story Points:** 5  
**Notes:**

* Ensure no overlapping with already scheduled appointments.
* Allow doctors to set one-time or recurring unavailability.

**User Story 5**

**Title:**  
*As a doctor, I want to update my profile with specialization and contact information, so that patients have up-to-date information.*

**Acceptance Criteria:**

1. A profile editing form should be available to the doctor.
2. Fields should include specialization, phone number, email, and bio.
3. Changes should be saved to the database and reflected immediately.

**Priority:** Medium  
**Story Points:** 3  
**Notes:**

* Validate input formats (e.g., phone, email).
* Consider moderation/approval for certain sensitive changes.

**User Story 6**

**Title:**  
*As a doctor, I want to view the patient details for upcoming appointments, so that I can be prepared.*

**Acceptance Criteria:**

1. Doctors should be able to view patient profiles from their appointment calendar.
2. Details should include patient history, reason for visit, and past consultations.
3. Access should be restricted to appointments the doctor is assigned to.

**Priority:** High  
**Story Points:** 4  
**Notes:**

* Display patient details in a modal or side panel for quick access.
* Implement privacy safeguards to prevent unauthorized access.