Manual Test Cases

(Lidia Polyakov)

API Test Cases

API-TC-1: Get All Bookings

Objective: Verify that the GET /booking endpoint returns all bookings successfully
Pre-conditions: API is accessible
Test Steps:

1. Send GET request to /booking endpoint
2. Verify response status code is 200
3. Verify response body contains array of booking objects
4. Verify each booking object has correct date

Expected Results:

- Status code: 200
- Response contains array of bookings

API-TC-2: Get Booking Room by Valid ID

Objective: Verify that GET /booking/{id} returns specific booking details
Pre-conditions: API is accessible, Valid booking ID exists
Test Steps:

1. Send GET request to /booking/{valid_id}
2. Verify response status code is 200
3. Verify response contains booking room details with correct name, price and date

Expected Results:

- Status code: 200
- Response contains booking room with name, price and date

API-TC-3: Get Booking room by Invalid ID

Objective: Verify that GET /booking/{id} handles invalid booking ID appropriately
Pre-conditions: API is accessible
Test Steps:

1. Send GET request to /booking/999999 (non-existent ID)
2. Verify response status code is 404
3. Send GET request to /booking/invalid (non-numeric ID)
4. Verify response status code is 404

Expected Results:

- Status code: 404 for both requests
- Appropriate error message

API-TC-4: Create Booking with Valid Data

Objective: Verify that POST /booking creates new booking successfully
Pre-conditions: API is accessible
Test Steps:

1. Prepare valid booking data (roomid, firstname, lastname, totalprice, depositpaid, bookingdates)
2. Send POST request to /booking with valid data
3. Verify response status code is 200
4. Verify response contains bookingid and booking object
5. Verify created booking by GET request using returned ID

Expected Results:

- Status code: 200
- Response contains bookingid and booking details
- New booking retrievable by ID

API-TC-5: Create Booking with Missing Required Fields

Objective: Verify that POST /booking handles missing required fields appropriately
Pre-conditions: API is accessible
Test Steps:

1. Send POST request with missing firstname field
2. Send POST request with missing bookingdates
3. Send POST request with empty request body
4. Verify appropriate error responses

Expected Results:

- Status code: 400 or 500
- Error message indicating missing required fields

API-TC-6: Login to Admin Portal

Objective: Verify that an admin can successfully log in using valid credentials.
Pre-conditions: Admin user account exists with valid credentials (username and password). Admin portal API endpoint is accessible.
Test Steps:

1. Send a POST request to `/auth` with valid admin credentials in the body.
2. Verify the response returns a status code of **200 OK**.
3. Verify the response body contains a valid authentication token or session identifier.
4. Send GET request to /booking?lastname=TestLastName
5. Verify response contains only bookings with matching lastname

Expected Results:

- Admin is successfully authenticated.
- Status code: 200
- Response includes a valid token.

API-TC-7: Send Contact Message

Objective: Verify that `PUT /message/{id}` updates a contact message successfully
Pre-conditions: A valid contact message exists, API endpoint is accessible.
Test Steps:

1. Create a new contact message using POST /contact
2. Send the request with updated contact details
3. Verify response Status code is 200 OK

4. Verify response data (name, email, phone, subject, message) reflect the updated values
5. Verify booking details are updated by GET request

Expected Results:

- Status code: 200
- Contact message is updated correctly
- Updated data retrievable via GET

UI Test Cases

UI-TC-1: Display Booking Page

Objective: Verify that the booking page displays correctly with all required elements
Pre-conditions: Navigate to booking page
Test Steps:

1. Open application URL
2. Verify booking page is visible
3. Verify presence of all required elements
4. Verify presence of checkin and checkout date pickers
5. Verify presence of Book button
6. Verify room information is displayed
7. etc.

Expected Results:

- All form elements visible and properly labeled
- Room details displayed with image, description, and price
- Contact form is ready for user input and submit send successfully

---

UI-TC-2: Submit Booking Room Form

Objective: Verify that booking room form submits successfully with valid data
Pre-conditions: Checkin and checkout dates sets, booking form is displayed.
Test Steps:

1. Click “Reserve Now”
2. Fill firstname, lastname , email, phone

3. Fill phone field with "1234567890
4. Click “Reserve Now”
5. Verify success message or confirmation

Expected Results:

- Form submits without errors
- Success message displayed
- Booking confirmation shown with the right dates

UI-TC-3: Admin Login with Valid Credentials

Objective: Verify that admin can login successfully with correct credentials
Pre-conditions: Navigate to admin login page
Test Steps:

1. Enter username and password
2. Click Login button
3. Verify successful login to admin dashboard

Expected Results:

- Successful login to admin dashboard
- Admin interface displayed
- Logout option available

UI-TC-4: Admin Login with Invalid Credentials

Objective: Verify that admin login rejects invalid credentials
Pre-conditions: Navigate to admin login page
Test Steps:

1. Enter invalid username or password
2. Click Login button
3. Verify login failure and error message

Expected Results:

- Login fails
- Error message displayed
- User remains on login page

UI-TC-5: Admin Portal Display

Objective: Verify that admin dashboard displays booking information correctly
Pre-conditions: Successfully logged in as admin
Test Steps:

1. Verify admin pages (rooms, report, branding, messages)
2. Verify action buttons (edit, delete, view) are present and works as expected

Expected Results:

- Each page displayed with data
- Management actions available
- Portal is functional and responsive
- etc.