Figma web-link:
https://www.figma.com/design/cpDnrK8kcSxIqUsZ3GYT5X/CISC-327-Group40-Flight-Ticket-Booking-System?node-id=4-2&t=RKjoIhquYGycapn9-1

Requirement 1: Users shall be able to register for a new account

Test case name: User Registration
Objective: To verify that users can successfully register for an account
Action: In the user registration page, fill in valid user details (e.g email, password)
Assert: Check that the user is registered and redirected to the login page with a success message

Test case name: Email already registered
Objective: To verify that the system prevents registration when a email is already registered
Action: In the registration page, enter an email that has already been used to register an account
Assert: Check that the registration is rejected and an error message is displayed

Test case name: Password Mismatch
Objective: To verify that the system rejects mismatched passwords during registration
Action: In the user registration page, enter a password and confirm it with a different password
Assert: Check that the registration is rejected and an error message is displayed

Test case name: Missing Required Fields
Objective: To verify that registration fails if required fields are missing
Action: In the user registration page, leave one or more required fields blank
Assert: Check that the registration is rejected and an error message is displayed

Requirement 2: Users shall be able to log in to their existing accounts

Test case name: Valid Login
Objective: To verify that users can log in successfully with valid credentials
Action: Enter a valid email and password on the login page
Assert: Check that the user is logged in and redirected to the homepage/dashboard

Test case name: Invalid Password
Objective: To verify that the system rejects login attempts with an incorrect password
Action: Enter a valid email and an incorrect password
Assert: Check that the login is rejected and an appropriate error message is displayed

Test case name: Invalid Email
Objective: To verify that the system rejects the login attempts with an incorrect email
Action: Enter an invalid email and a valid password

Assert: Check that the login is rejected and an appropriate error message is displayed

Test case name: Empty Email or Password
Objective: To verify that the login process does not proceed with empty input fields
Action: Leave the email or password field blank and try to log in
Assert: Check that the login attempt is rejected, and error messages are displayed for empty fields

Requirement 3: **Users should be able to view their account information and make changes**

Test case name: View Profile Information
Objective: To verify that users can view their profile information after logging in
Action: Log in and navigate to the profile management page
Assert: Check that all the user's profile details are correctly displayed

Test case name: Update Profile Information
Objective: To verify that users can successfully update their profile information
Action: Navigate to the profile management page and update personal details
Assert: Check that the profile information is updated, and a confirmation message is displayed

Requirement 4: **Users shall be able to search for available flights by entering details such as departure city, destination city, travel dates, and number of passengers.**

Test Case Name: Flight Search
Objective: To verify that users can search for flights based on the entered criteria.
Action: On the flight search page, enter valid departure city, destination city, travel dates, and number of passengers, then click on the "Search" button.
Assert: Ensure that a list of available flights matching the criteria is displayed.

Test Case Name: No Flights Available
Objective: To verify the system behaviour when no flights match the search criteria.
Action: Enter departure city, destination, travel dates, and number of passengers that do not match any flights, then click "Search."
Assert: Ensure that a "No flights available" message is shown.

Test Case Name: Invalid Search Input
Objective: To check that the system handles invalid search inputs.
Action: Enter incomplete or invalid data (ex: missing destination city or invalid date format) and click “Search”.
Assert: Ensure that an error message is displayed, and the user is prompted to correct the input

Test Case Name: Return Flight Search

Objective: To verify that users can search for return flights by entering valid departure and return details.
Action: On the flight search page, select "Round Trip," then enter valid departure and destination cities, departure date, return date, and number of passengers, and click "Search."
Assert: Ensure that two lists of available flights are displayed—one for outbound and one for return flights.

Test Case Name: Invalid Return Date
Objective: To check if the system handles invalid return date entries (e.g., return date earlier than departure date).
Action: Enter a valid departure date and an invalid return date (before the departure date), then click "Search."
Assert: Ensure that an error message is shown, and the user is prompted to correct the return date.

Requirement 5: Users shall be able to select seats for a chosen flight from the available seating options.

Test Case Name: Seat Selection
Objective: To verify that users can successfully select a seat.
Action: After choosing a flight, navigate to the seat selection page. Click on an available seat and confirm selection.
Assert: Ensure that the selected seat is highlighted, and the seat number is shown in the booking summary.

Test Case Name: Already Booked Seat
Objective: To verify the system behaviour when a user tries to select an already booked seat.
Action: Try to click on a seat that is marked as booked.
Assert: Ensure that an error message is shown, and the user is prevented from selecting the booked seat.

Test Case Name: No Seat Selected
Objective: To check if the user is prompted when no seat is selected before confirming the booking.
Action: Attempt to proceed to payment without selecting a seat.
Assert: Ensure that the user is shown an error message instructing them to select a seat before proceeding.

Requirement 6: Users shall be able to make payments using a form with fields for credit card details, billing address, etc.

Test Case Name: Payment Form Validation
Objective: To verify that the payment form accepts valid credit card details and billing information.
Action: On the payment page, enter valid credit card information and billing details, then submit the form.

Assert: Ensure that the payment is successfully processed, and the user is redirected to a confirmation page.

Test Case Name: Invalid Credit Card
Objective: To verify that the system handles invalid credit card details.
Action: Enter incorrect credit card information (e.g., wrong card number or expired card) and submit the form.
Assert: Ensure that an error message is displayed, and the payment is not processed.

Test Case Name: Required Fields
Objective: To ensure that all required fields must be filled out for successful payment.
Action: Try to submit the payment form with one or more fields (e.g., card number or billing address) left blank.
Assert: Ensure that the user is shown an error message, and they are required to complete all mandatory fields.

Requirement 7: **Users should be able to view their previous flight bookings**

Test Case Name: Access View Booking History
Objective: To make sure that users can view their previously booked tickets
Action: Log into the user account, navigate to the Booking History page
Assert: Check that a list of previously booked flights is displayed, showing relevant details such as flight number, date, time and status

Test Case Name: No Previous Bookings
Objective: To make sure that the system displays appropriate message when no flight bookings are found
Action: Log into a user account with no previous flight bookings and navigate to the Booking History page
Assert: Check that a message like “No previous bookings” is displayed

Test Case Name: Booking History Error Handling
Objective: Verify that the system handles errors properly when the booking history cannot be retrieved
Action: Simulate a system error (e.g., server outage or API failure) and attempt to view the "Booking History" page.
Assert: Check that an error message like “Unable to retrieve booking history” Is displayed

Requirement 8: **The flight booking system should provide customer support**

Test Case Name: Access Customer Support
Objective: To verify that users can successfully access the customer support section of the flight booking system.
Action: Log into the user account and navigate to the "Customer Support" or "Help Center" page from the main menu.

Assert: Check that the customer support page is displayed with options such as FAQ, live chat, email support, and contact numbers.

Test Case Name: Submit a Support Ticket
Objective: To verify that users can submit a support ticket for assistance.
Action: In the "Customer Support" page, fill in a support ticket form with valid details (e.g., issue description, contact info) and submit.
Assert: Check that the support ticket is successfully submitted, and the user receives a confirmation message like "Your request has been submitted."

Test Case Name: Customer Support Contact Information
Objective: To verify that customer support contact details (phone number, email) are available for users.
Action: Navigate to the "Customer Support" or "Contact Us" section of the website.
Assert: Check that the system displays the correct customer support phone numbers and email addresses.

Test Case Name: FAQ Section Availability
Objective: To verify that the FAQ section is available and provides relevant answers.
Action: Navigate to the "Customer Support" or "Help Center" page and access the FAQ section.
Assert: Check that the FAQ section contains common questions with answers related to flight bookings, cancellations, and payment issues.

Requirement 9: Users should be able to refund and cancel plane tickets

Test Case Name: Cancel a Ticket
Objective: To verify that users can successfully cancel their booked flight tickets.
Action: Log into the user account, navigate to the "My Bookings" or "Booking History" page, select a booked ticket, and choose the "Cancel" option.
Assert: Check that the ticket is successfully canceled, and the user receives a confirmation message such as "Your booking has been canceled" along with relevant details (e.g., refund policy, fees).

Test Case Name: Cancellation and Refund Eligibility
Objective: To verify that the system correctly identifies tickets that are not eligible for cancellation or refund (e.g., non-refundable tickets, past flight dates).
Action: Attempt to cancel or request a refund for a non-refundable or expired ticket.
Assert: Check that the system displays an error or information message like "This ticket is not eligible for cancellation or refund," preventing the user from proceeding with the action.

Test Case Name: Refund Processing Status
Objective: To verify that users can track the status of their refund request after cancellation.
Action: Log into the user account, navigate to the "My Bookings" or "Refund History" page, and check the status of a previously submitted refund request.
Assert: Check that the system displays the current status of the refund (e.g., "Pending," "Approved," "Processed") along with an estimated timeline for completion and any additional relevant details.