

```

test('Homepage loads with all necessary elements', async () => {
  await page.goto('https://yourflightbookingapp.com');
  await expect(page).toMatchElement('header');
  await expect(page).toMatchElement('nav');
  await expect(page).toMatchElement('#search-form');
  await expect(page).toMatchElement('footer');
});

```

```

test('Flight search form renders with all required fields', async ()
  => {
  await page.goto('https://yourflightbookingapp.com');
  await expect(page).toMatchElement('#from-input');
  await expect(page).toMatchElement('#to-input');
  await expect(page).toMatchElement('#departure-date');
  await expect(page).toMatchElement('#return-date');
  await expect(page).toMatchElement('#search-button');
});

```

```

it('Searches for direct flights with valid inputs', () => {
  cy.visit('https://yourflightbookingapp.com');
  cy.get('#from-input').type('New York');
  cy.get('#to-input').type('Los Angeles');
  cy.get('#search-button').click();
  cy.get('.flight-result').should('exist');
});

```

```

it('Searches for multi-stop flights with valid inputs', () => {
  cy.visit('https://yourflightbookingapp.com');
  cy.get('#from-input').type('New York');
  cy.get('#to-input').type('Tokyo');
  cy.get('#stops-select').select('1 Stop');
  cy.get('#search-button').click();
  cy.get('.flight-result').should('exist');
});

```

```

test('Retrieves weekly direct flights matching current week schedule'
  , () => {
  const flights = getWeeklyFlights('NYC', 'LAX', 'direct');
  expect(flights).toHaveLength(7); // Assuming 1 flight per day for
    simplicity
});

```

```

test('Calculates correct total flight time for direct flights', () =>
  {
  const totalTime = calculateTotalFlightTime('09:00', '15:00'); //
    Using HH:MM format
  expect(totalTime).toBe('6 hours');
});

```

```
test('Converts 12-hour format to 24-hour format accurately', () => {
  const time24 = convertTo24Hour('2:00 PM');
  expect(time24).toBe('14:00');
});
```

```
test('Converts 24-hour format to 12-hour format accurately', () => {
  const time12 = convertTo12Hour('14:00');
  expect(time12).toBe('2:00 PM');
});
```

```
test('Logs in successfully with correct credentials', async () => {
  await page.goto('https://yourflightbookingapp.com/login');
  await page.type('#username', 'validUser');
  await page.type('#password', 'validPassword');
  await page.click('#login-button');
  await page.waitForNavigation();
  expect(page.url()).toBe('https://yourflightbookingapp.com/dashboard');
});
```

```
test('Fails to log in with incorrect credentials', async () => {
  await page.goto('https://yourflightbookingapp.com/login');
  await page.type('#username', 'invalidUser');
  await page.type('#password', 'invalidPassword');
  await page.click('#login-button');
  await expect(page).toMatch('Invalid username or password');
});
```

```
test('Fetches correct flights based on search criteria', async () => {
  const searchCriteria = { from: 'NYC', to: 'LAX', date: '2023-10-05' };
  const results = await fetchFlights(searchCriteria);
  expect(results).toBeInstanceOf(Array);
  results.forEach(flight => {
    expect(flight.from).toBe('NYC');
    expect(flight.to).toBe('LAX');
    // Assuming date comparison logic or library (e.g., moment.js) for simplicity
    expect(new Date(flight.departureDate)).toEqual(new Date('2023-10-05'));
  });
});
```

```
test('Saves booking details correctly', async () => {
  const bookingDetails = { flightId: '123', userId: '456', seats: 2,
    status: 'confirmed' };
  const savedDetails = await saveBookingDetails(bookingDetails);
  expect(savedDetails).toMatchObject(bookingDetails);
});
```

```
test('Processes payment successfully with valid details', async () => {
  {
    const paymentDetails = { cardNumber: '4111111111111111', expiry:
      '12/24', cvv: '123', amount: 100 };
    const status = await processPayment(paymentDetails);
    expect(status).toBe('success');
  }
});
```

```
test('Fails to process payment with invalid details', async () => {
  const paymentDetails = { cardNumber: '4111111111111112', expiry:
    '12/24', cvv: '123', amount: 100 };
  const status = await processPayment(paymentDetails);
  expect(status).toBe('failure');
});
```

```
test('Displays user-friendly error messages for common errors', async
  () => {
    await page.goto('https://yourflightbookingapp.com/error-prone
      -operation');
    await page.click('#trigger-error');
    await expect(page).toMatchElement('.error-message', { text:
      'Something went wrong, please try again.' });
  });
```

```
jest.mock('../services/errorLoggingService');

test('Logs errors correctly for monitoring and debugging', () => {
  const errorToLog = new Error('Test error');
  handleOperationThatMayFail(); // This function should internally
    call logError on failure
  expect(logError).toHaveBeenCalledWith(errorToLog);
});
```