

# Group 15 CISC 327 - Fall 2024

## Team Members:

- Youssef Elmanawy
  - Aidan Sibley
  - Will Wu
- 

## Project Overview

Welcome to the Group 15 project for CISC 327 - Fall 2024. This project is a web application platform for booking flights. The system allows users to search for flights, select options like passengers, view search results, and perform other booking-related activities. The goal is to provide users with a smooth and enhanced booking experience.

This README provides a comprehensive guide to set up the project on your local machine, navigate the project directory, understand the file structure, and run the tests.

---

## Directory Structure Overview

```
.
├── Assignment-1
│   └── group15-assignment1.pdf          # Previous assignment files
├── Assignment-2
│   ├── readme-first.md                 # Instructions specific to Assignment-2
│   ├── screenshots                     # Screenshots of the test results
│   └── task-dist.md                    # Contribution details
├── pnpm-lock.yaml                      # Lockfile for pnpm dependencies
├── project-ideas.md                    # Initial project ideas and brainstorming document
├── README.md                           # The main README file (you are here)
├── src                                 # Main application directory
│   ├── components                      # All application components
│   │   ├── pages
│   │   │   ├── landing                # Components for the landing page
│   │   │   │   ├── CTA
│   │   │   │   │   ├── DropdownSelect.tsx
│   │   │   │   │   ├── PassSelect.tsx
│   │   │   │   │   └── __tests__
│   │   │   │   │       ├── DropdownSelect.test.tsx
│   │   │   │   │       ├── FlightSearchForm.test.tsx
│   │   │   │   │       └── PassSelect.test.tsx
│   │   └── search-results              # Components for search results page
│   │       ├── components
│   │       │   ├── DateTabs.tsx
│   │       │   ├── FlightCard.tsx
│   │       │   ├── FlightsList.tsx
│   │       │   ├── LoadMoreButton.tsx
│   │       │   ├── InfoBanner.tsx
│   │       │   └── SearchHeader.tsx
│   │       ├── index.tsx               # Main page for search results
│   │       ├── service.ts              # Mock service to retrieve flight data
│   │       └── __tests__
│   │           ├── DateTabs.test.tsx
│   │           ├── FlightCard.test.tsx
│   │           ├── FlightsList.test.tsx
│   │           ├── LoadMoreButton.test.tsx
│   │           └── SearchHeader.test.tsx
```

jest.config.ts	# Jest configuration for running tests
jest.setup.ts	# Jest setup file
next.config.mjs	# Next.js configuration
package.json	# Project dependencies and scripts managed by pnpm
public	# Static assets and dummy data
dummy_data	
destinations.json	# Dummy data for destinations
flights.json	# Dummy data for flights
styles	# Global and base styles for the front-end
tsconfig.json	# TypeScript configuration

---

## Setup Instructions

### Prerequisites

Before you begin, ensure you have the following installed on your machine: - **Node.js 16+**: JavaScript runtime for the front-end (Next.js). - **pnpm**: Package manager for the front-end. Install via: `npm install -g pnpm`

### Steps to Get Started

**1. Clone the Repository** First, clone the repository to your local machine:

```
git clone https://github.com/ajsib/Group15-CISC327-F24.git
cd project
```

**2. Install Dependencies** Navigate to the project directory and install all the required dependencies:

```
cd src
pnpm install
```

**3. Run the Development Server** To start the development server, run:

```
pnpm dev
```

The Next.js server should now be running on `http://localhost:3000/`.

### Running the Tests

**Running All Tests:** To run the test suite and ensure everything works correctly, run:

```
pnpm test
```

This will execute all the unit tests across the project. The test results, including passing and failing tests, will be displayed in the terminal.

### Available Test Files

Here is a breakdown of the test files available in the project:

- **Landing Page Tests:**
  - `DropdownSelect.test.tsx`: Tests for the dropdown selection component.
  - `FlightSearchForm.test.tsx`: Tests the form where users input flight search criteria.
  - `PassSelect.test.tsx`: Tests passenger selection functionality.
- **Search Results Page Tests:**
  - `DateTabs.test.tsx`: Tests for the date tab selection component.
  - `FlightCard.test.tsx`: Tests for the flight card component that displays flight details.
  - `FlightsList.test.tsx`: Ensures the list of flights is correctly rendered.
  - `LoadMoreButton.test.tsx`: Verifies the "Load More" button behavior.
  - `SearchHeader.test.tsx`: Tests the search header that displays search criteria.

Each test includes cases for rendering, user interactions, and correct data display.

### **Mock Data for Testing**

- **Flights:** The flight data is stored in `public/dummy_data/flights.json`.
- **Destinations:** The destinations data is in `public/dummy_data/destinations.json`.

These JSON files contain dummy data used to mock the flight search functionality, allowing the application to run without connecting to a real backend.