Tutorial + Task Distribution + File Structure

TUTORIAL
Running the program

Ensure that both frontend(Reactjs) and backend(Flask) running.

Frontend

Prerequisites

Before you begin, make sure you have the following installed:
- Node.js (version 18 or higher)
- npm (comes with Node.js)

Installation to work with (Front end)

1. Clone the repository:
2. cd your-repo-name/frontend
3. npm install
4. npm run dev

Backend

Installation to work with (back end)

To set up and run the Flask backend, follow these steps:

Navigate to the backend directory

cd ../backend

Create a virtual environment:

python3 -m venv venv

Activate the virtual environment

On Linux or macOS: source venv/bin/activate On Window: venv\Scripts\activate

Install Flask Dependencies

pip install -r requirements.txt

Set Up the Flask Application

Set the Flask application environment variable to point to the main Flask file (app.py): **export FLASK_APP=app.py**

Run the Flask Server

flask run

How to run Test
We used react testing library
Make sure you are in the correct directory:
1. Assume you are in the project's root directory:

   cd src/frontend

2. To run test cases:
   a. Install dependencies: (this includes the necessary libraries to run the project as well as run the test cases)
      npm install

   b. Run test cases:
      npm test

Successful run of all test cases should display as following in your terminal:

```
√ src/__test__/TestBookingPayment.test.jsx (4)
√ src/__test__/TestFlightBooking.test.jsx (4)
√ src/__test__/TestNavBar.test.jsx (3)
√ src/__test__/TestRegister.test.jsx (5)
√ src/__test__/TestSignIn.test.jsx (4)

 Test Files  5 passed (5)
      Tests  20 passed (20)
   Start at  19:57:08
   Duration  1.57s (transform 338ms, setup 0ms, collect 2.52s, tests 878ms, environment 2.79s, prepare 417ms)
```
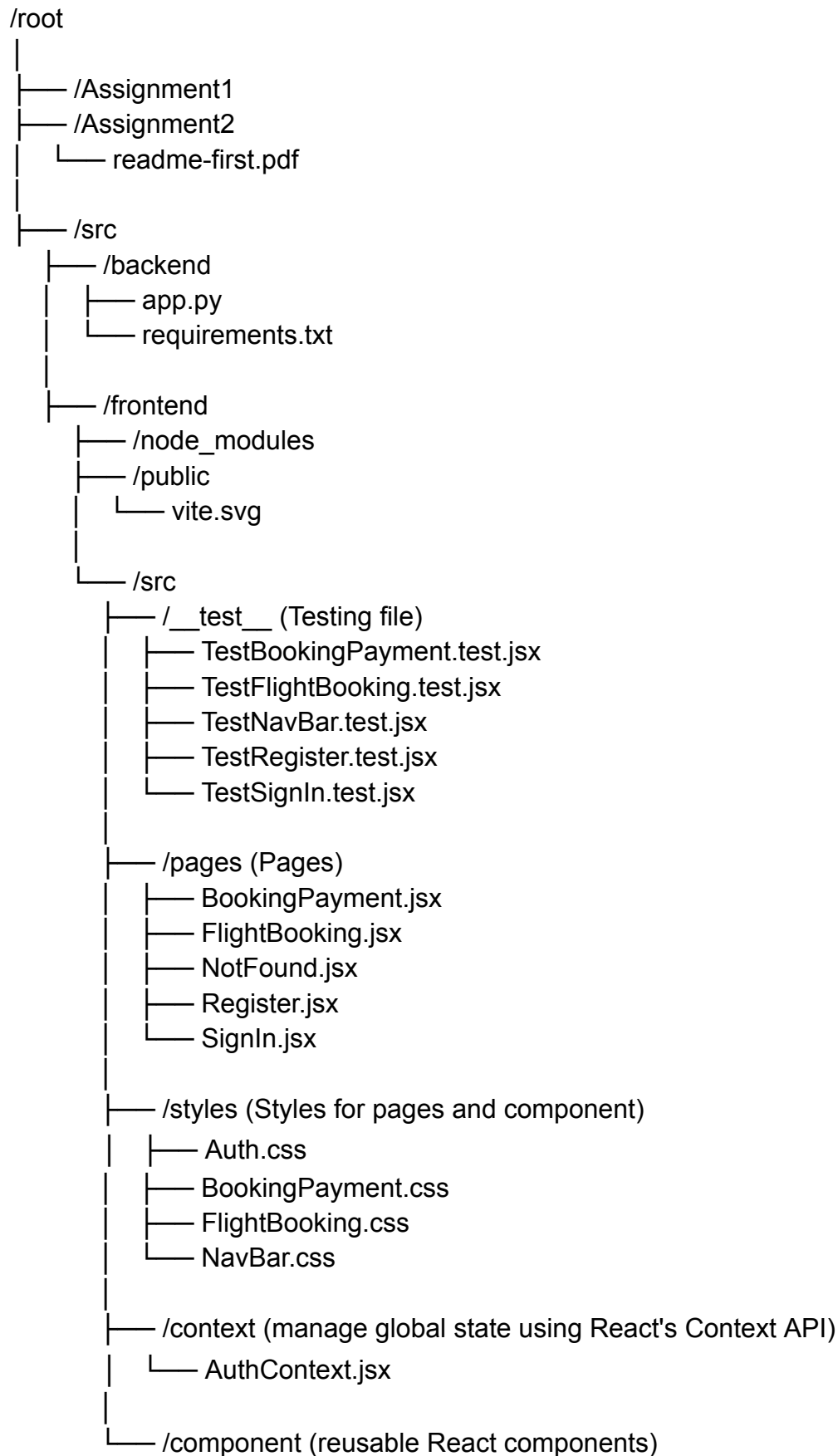
All test files with extension ".test.jsx" are under "src/__test__" directory, each file corresponds to a component

TASK DISTRIBUTION

- Gia Nguyen (giahoangnguyen):
  - Implement booking payment feature for homepage (.jsx + css style)
  - Write unit test for booking payment feature
  - Implement file structure tree
  - Write readme-first template
- Vu Thanh Loc Mai (locmai27):
  - Implement flight search feature for home page
  - Set up test libraries for front end
  - Write unit test for homepage rendering and search feature
  - Set up backend template
  - Write readme-first template
- Nguyen Nguyen (Nguyen1611):
  - Set up signIn, Register page
  - Set up Navbar (in component folder) and routes to pages
  - Write unit test for Navbar, SignIn, Register
  - Set up CORS to connect backend and frontend
  - Write readme-first template

BELOW IS FILE STRUCTURE WITH EXPLAINED FOLDER

FILE STRUCTURE

```
/root
│
├── /Assignment1
├── /Assignment2
│    └── readme-first.pdf
│
├── /src
│    ├── /backend
│    │    ├── app.py
│    │    └── requirements.txt
│    │
│    ├── /frontend
│         ├── /node_modules
│         ├── /public
│         │    └── vite.svg
│         │
│         └── /src
│              ├── /__test__ (Testing file)
│              │    ├── TestBookingPayment.test.jsx
│              │    ├── TestFlightBooking.test.jsx
│              │    ├── TestNavBar.test.jsx
│              │    ├── TestRegister.test.jsx
│              │    └── TestSignIn.test.jsx
│              │
│              ├── /pages (Pages)
│              │    ├── BookingPayment.jsx
│              │    ├── FlightBooking.jsx
│              │    ├── NotFound.jsx
│              │    ├── Register.jsx
│              │    └── SignIn.jsx
│              │
│              ├── /styles (Styles for pages and component)
│              │    ├── Auth.css
│              │    ├── BookingPayment.css
│              │    ├── FlightBooking.css
│              │    └── NavBar.css
│              │
│              ├── /context (manage global state using React's Context API)
│              │    └── AuthContext.jsx
│              │
│              └── /component (reusable React components)
```

└── Navbar.jsx