

Tutorial + Task Distribution + explanation Integration test

TUTORIAL

Github:

<https://github.com/Nguyen1611/CISC327-QA-Project-Group-60-AA>

Please create a .env file in src/.env (same directory level as backend and frontend folder) add the DATABASE_URI

`DATABASE_URI=mongodb+srv://22rdkr:pn4ai8c8bVBTjPmj@cisc327-project.jm2de.mongodb.net/`

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 End To End Test (Cypress)

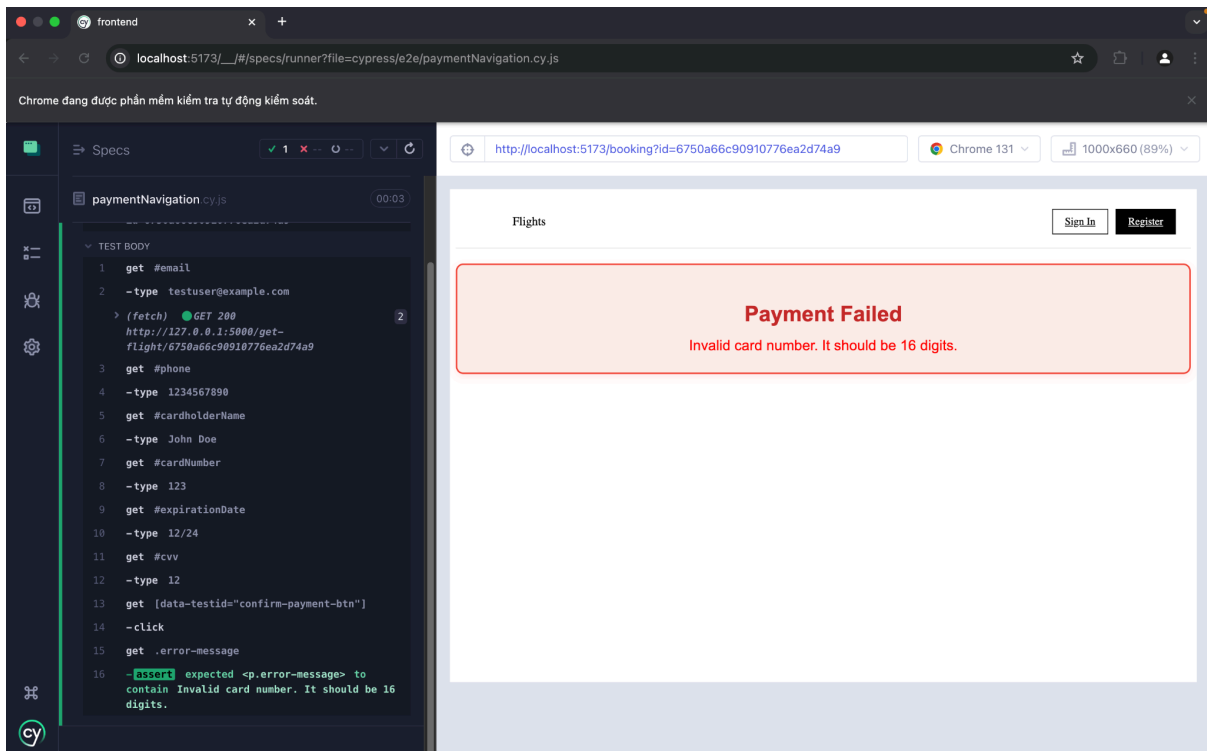
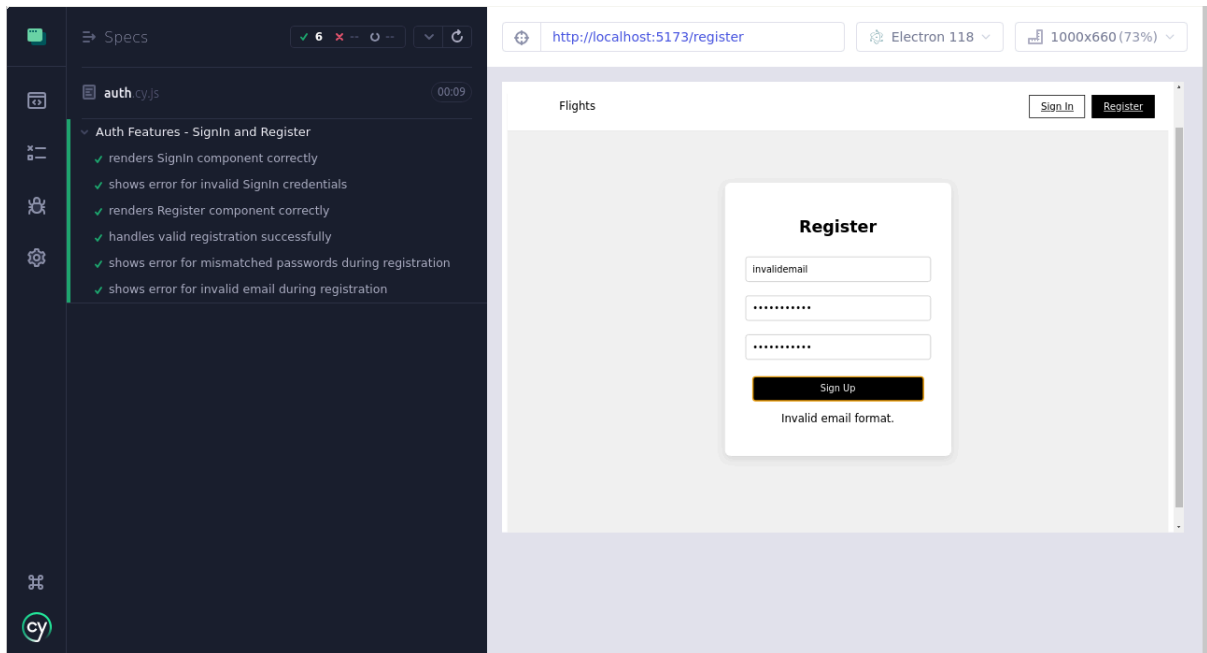
Make sure to open the backend and frontend server and set up database uri .env properly

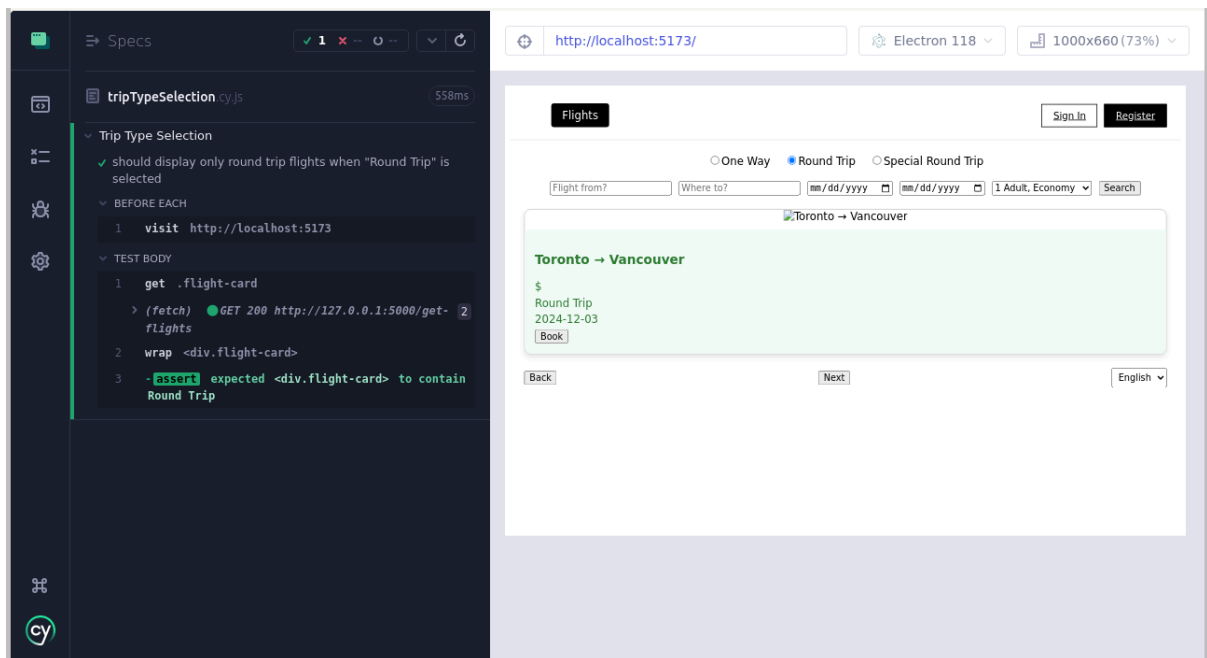
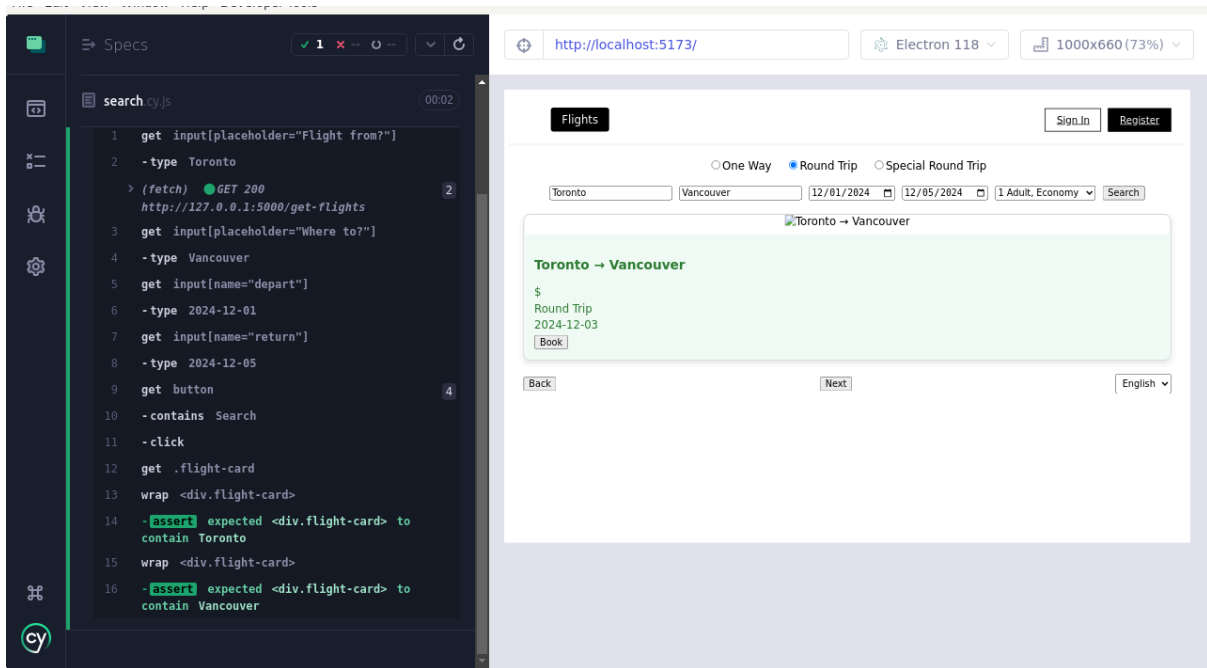
cd src/frontend

```
npm install
```

```
sudo npx cypress open
```

END TO END SCREENSHOT





E2E TEST EXPLANATION:

1. Flight Search: Ensure the search filters flights correctly based on departure and destination locations, trip type, and date range.
2. Flight Booking Navigation: Ensure the "Book" button navigates to the correct booking page for a selected flight.
3. Trip Type Selection: Ensure the flight results change dynamically when different trip types are selected.
4. Booking Payment Navigation: The test verifies the **BookingPayment component** functionality in a flight booking application. It starts by navigating to a booking page using a

valid flight ID (<http://localhost:5173/booking?id=6750a66c90910776ea2d74a9>) and ensures the flight details are displayed correctly. The user inputs payment and contact information, confirms the payment, and checks if the application redirects to the "Payment Successfully" page upon successful booking. This ensures the component integrates properly with the backend and handles user input, validation, and navigation as expected.

5) Authentication: . The tests validate the proper rendering of input fields and buttons, and ensure functionality for user login and registration. For SignIn, tests cover successful and failed login attempts with appropriate error messages. For Register, tests validate successful registration, mismatched passwords, and invalid email formats, along with proper error handling.

TASK DISTRIBUTION

- Nguyen Nguyen:
 - write e2e test script for Signin and Register features
- Loc Mai:
 - Set up cypress, write e2e test scripts for Search feature
- Gia Nguyen
 - write e2e test script for Booking Payment feature