Tutorial + Task Distribution + explanation missed lines

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/
```

How to run Coverage

Frontend coverage:

```
cd src/frontend
npm install
npm run coverage
```

Backend Coverage

Backend Coverage:

```
cd src/backend
```

```
python3 -m venv venv
```

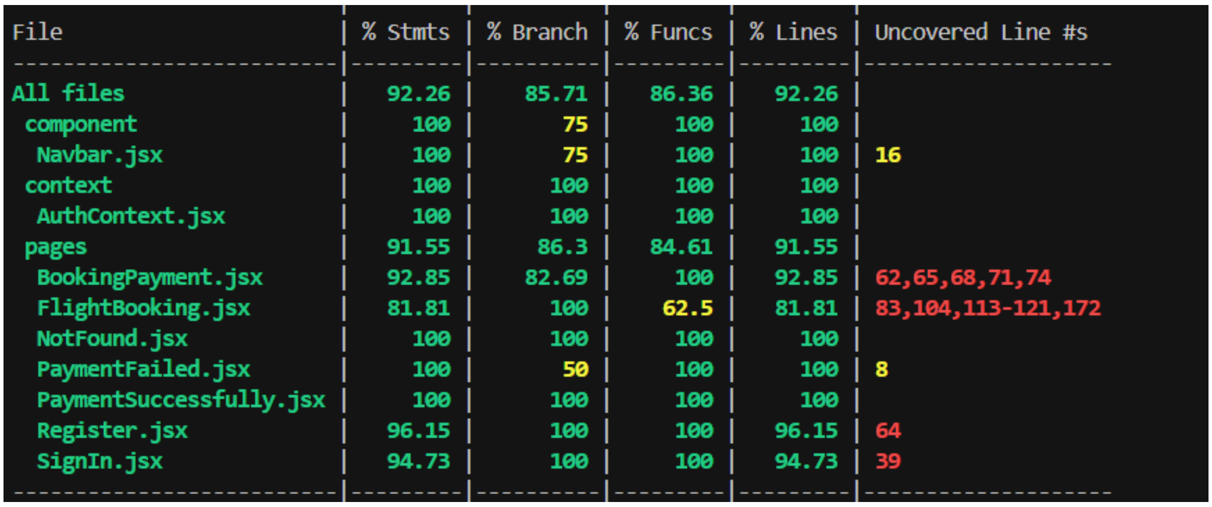
Activate the virtual environment

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

```
pip install -r requirements.txt
```

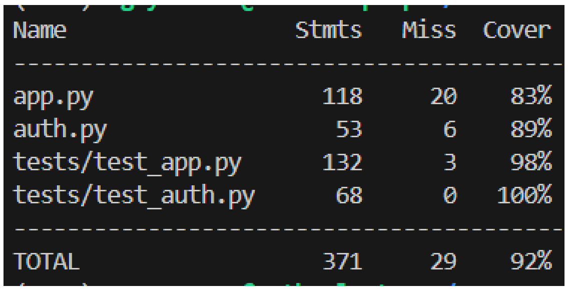
```
coverage run -m pytest
coverage report -m
```

Frontend Screenshot:



```
File                      | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
--------------------------|---------|----------|---------|---------|-------------------
All files                 |   92.26 |    85.71 |   86.36 |   92.26 |
 component                |     100 |       75 |     100 |     100 |
  Navbar.jsx              |     100 |       75 |     100 |     100 | 16
 context                  |     100 |      100 |     100 |     100 |
  AuthContext.jsx         |     100 |      100 |     100 |     100 |
 pages                    |   91.55 |     86.3 |   84.61 |   91.55 |
  BookingPayment.jsx      |   92.85 |    82.69 |     100 |   92.85 | 62,65,68,71,74
  FlightBooking.jsx       |   81.81 |      100 |    62.5 |   81.81 | 83,104,113-121,172
  NotFound.jsx            |     100 |      100 |     100 |     100 |
  PaymentFailed.jsx       |     100 |       50 |     100 |     100 | 8
  PaymentSuccessfully.jsx |     100 |      100 |     100 |     100 |
  Register.jsx            |   96.15 |      100 |     100 |   96.15 | 64
  SignIn.jsx              |   94.73 |      100 |     100 |   94.73 | 39
--------------------------|---------|----------|---------|---------|-------------------
```

Backend Screenshot:



```
Name                    Stmts   Miss  Cover
-------------------------------------------
app.py                    118     20    83%
auth.py                    53      6    89%
tests/test_app.py         132      3    98%
tests/test_auth.py         68      0   100%
-------------------------------------------
TOTAL                     371     29    92%
```

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

FRONT-END:

- for FlightBooking.jsx:
    - there is currently an error when testing useNavigate() function, I think this might be due to a conflict. It said that useNavigate() should only be used under Router context and useNaviate() is inside a Route component which is already defined under Router in App.jsx
    - for line 83, I cannot simulate a network error
    - for line 104, since useNavigate() is identified as an error in the previous line, this line is automatically deduced as uncovered by the testing library
    - for line 113-121, since other filters are tested (‘Special Round Trip’), I felt like it is redundant to test all the filters for this search.
    - for line 172, this button triggers the useNavigate() function to navigate to route ‘/booking’ so it is also deduced as uncovered automatically by the testing library

- for BookingPayment.jsx: lines 62,65,68,71,74
    - These lines are not covered because all of them are about testing for invalid payment. The reason for not writing a test case for the validatePaymentInfo function itself is that the primary logic of this function is already covered indirectly through the user flow and navigation behavior in the component. Specifically, when the payment information is invalid, the validatePaymentInfo function sets an error message, and the user is redirected to the PaymentFailed component. This redirection and the error handling are already tested by checking the navigation behavior when the payment fails. Since the invalid payment scenario triggers navigation to a different component (PaymentFailed), and this navigation flow is already validated with existing tests, there is no need to write separate tests for the validatePaymentInfo function. Essentially, the focus of testing in this case is on the overall user experience and ensuring that the navigation occurs correctly when the payment fails, which inherently validates the function's behavior without needing direct unit tests for the validation logic itself.
- for Signin.jsx:
    - I feel like testing for the network error case is not needed as in my code when there is network error, the sign in process does not work
- For Register.jsx:
    - There is no code at that line

BACK-END: lines 149-151, 214-215, 226

- for test_app.py: All the uncovered lines are lines about testing whether mongoDB is connected successfully or not, as well as connecting to the userdatabase and flightdatabase successfully or not. However, the above tests can all be performed by connecting directly to the web, in addition, we also have a condition about testing whether mongoDB is connected successfully or not inside app.py, which is Therefore, testing the above lines will not be necessary in code coverage.

- for test_auth: Most of the lines are empty space or initialize mock details for testing

TASK DISTRIBUTION
- Nguyen Nguyen:
    - Write backend test script for coverage for auth.py
    - Write frontend test script for coverage for signin,register,notfound,authContext
- Loc Mai:
    - write additional frontend test for coverage for FlightBooking.test.jsx
    - write additional backend test for coverage for ‘/get-flights’ and ‘/get-flight/<id>’ endpoints
- Gia Nguyen
  - Write additional frontend test for coverage for BookingPayment.test.jsx

- Write backend tests for function test_getBookingHistory, test_confirmBooking, test_before_request, test_is_valid_payment, test_users_collection.
-