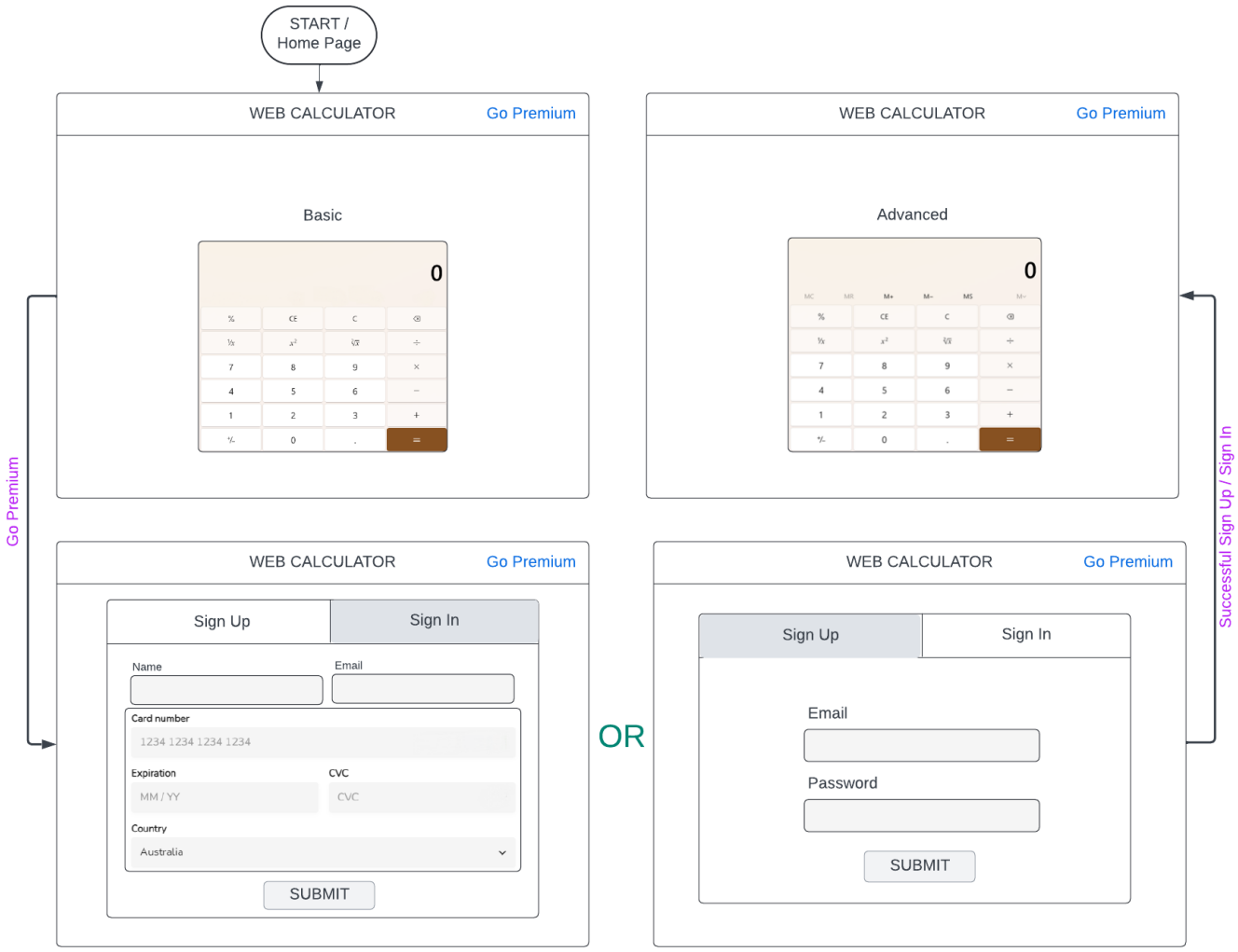
Programming Challenge for
Full-Stack software developer at Funlab

Build a Web Calculator



Desired Pages, Process Flow and Look-n-Feel

The task is to build a web calculator with following features:

1. A basic calculator in the home page
2. An option to “Go Premium”
3. **Stripe** payment page when user goes premium.
4. Premium users get to use some advanced features of calculator.

Platform requirements:

1. Frontend – React Typescript
2. Backend – ASP.Net Core (C#)
3. DB – SQL Server or JSON file

Design requirements:

1. Home page will have the basic calculator with “Go Premium” link at top right corner.
2. All functionalities of calculator should be done in frontend.
3. Clicking “Go Premium” will open an overlayed pop-up window with two tabs:
    a. ***Sign Up***: it should look like the bottom left page. When a user submits the page, it should send all the info to backend and backend needs to validate the credit card info via Stripe API. Once validated and Stripe payment is successful, *a welcome email is sent to users email with an auto-generated password*. For all successful sign-ups, the name, email, password, last four digit of card number and expiry date should be stored in DB or JSON file. For a successful sign-up, it should show the Advanced calculator page like top right one. For unsuccessful sign-up attempts, show an appropriate error message as a pop-up.
    b. ***Sign In***: It should look like the bottom right page. Once the page is submitted, the information is sent to backend and backend validates the email and password against the DB or JSON file. For a successful sign-in, it should show the Advanced calculator page like top right one. For unsuccessful sign-in attempts, show an appropriate error message as a pop-up.
4. Communication with Stripe must be done from backend.
5. Premium users can access some extra functionalities in the calculator (those memory operations as indicated in the top right page).
6. Stripe payment can be done using Stripe’s **Test Visa Card**. Successful sign-up will charge $10 from the test card.

Optional (but has a bonus):

1. Instead of a one-time payment, Stripe payment will create a monthly subscription charging $5 / month starting from the day of signing up.
2. Unit tests for frontend and backend.
    a. Each calculator operations should be performed in an individual functions which should have unit tests.
    b. For backend, each Stripe operation should be inside a separate function so it can be unit tested. Each operation with DB / JSON file should be inside separate function as well for the same reason.

Submission procedure:

1. The complete project should be hosted in Github and the link should be provided to us.
2. The stripe account used for this assignment should be provided to us (the credentials), so that we can verify the transactions that are happening. It would be ideal to create a separate Stripe account for this project.