Victoria Guardian Front-end Project

This is the front-end codebase of Victoria Guardian. The codebase is stored in a private repository on Github. If the permission is granted, users are authorized to fork the repository and run their own codebase.

This project is built in React along with the state management library Redux. React router is introduced to manage the application routing. Styled component is used to manage the styling of components. Following the TDD paradigm, the codebase covers a range of testing. For example, we create unit tests for various functions and APIs, the UI components are tested on the Storybook which allows you to develop UI components in isolation. In addition, integration testing is also set up to ensure the quality of the code before deployment.

Live website: https://invasivespecies-500c6.appspot.com

Password: fit5120

Installation

Porject environment

- Node: v9.11.1
- Create-react-app: v1.4.3
- React: v16.2.0
- Yarn: v1.6.0
- NPM: v5.6.0

Run the code on a local machine

```
npm install
npm start
open http://localhost:3000
```

If use yarn

```
yarn install
yarn start
open http://localhost:3000
```

Testing

Testing reliability and robustness of the application is very important for ensuring the quality of system. Our project adopts the TDD(Test Driven Development) paradigm. Our testing covers unit tests, integration tests and usability tests.

Run unit tests

```
npm test
```

This command will execute all the test cases in our codebase. The test cases include testing utility tools and APIs used in our project.

Run Storybook

Storybook runs outside of the application. It enables us build the UI components quickly without worrying about the dependencies. Learn more...

```
npm run storybook
open http://localhost:9009
```

Storybook Screenshot The component list are on the left. The components are showcased on the right hand side.

We can look and feel the UI components easily on the panel. Any change we made in the component will be shown on Storybook immediately.

Integration testing

The project also adopts integration testing by using Travis CI. Travis CI is a hosted, distributed continuous integration service used to build and test software projects hosted at GitHub. Learn more...

Everytime when the changes are committed and pushed to the remote repository, Travis CI will be activated and run the test suites in the project automatically.

Travis CI is configured by adding a file named .travis.yml, which is a YAML format text file, to the root directory of the repository.

Travis CI creates a log for the building and testing process. Travis CI screenshot

---

Deploying on Google Cloud Platform

This project currently is hosted on Google Cloud Platform. The project ID is **invasivespecies-500c6**

Deployoment Steps

1. In order to deploy the codebase to Google Cloud, it is required to install the Google Cloud SDK first. Learn more...

2. After you install Cloud SDK, the next step is typically run the gcloud init command to perform initial setup tasks. You can also run gcloud init at a later time to change your settings or create a new configuration. During the initialisation process, you are also required to complete the authorization step when prompted. Learn more...

```
gcloud init
```

3. Configure the deployment setting. Basically you just need to add the configurattion file **app.yaml** to the root directory. Learn more...

4. Deploy onto the Google cloud: execute the command

   ```
   gcloud app deploy --project your-project-id
   ```

   The app engine will be deployed in several minutes. You can then browse the website with the command

   ```
   gclound app browse
   ```

Administration

Once the project is deloyed, you can log on Google Cloud Platform to check out the status of the project. The dashboard on the front page shows the overview of the current project including the traffic of the website and the numbers of requests in a period of time. Dashboard

App Engine application is auto-scalable which means that if the traffic increases, the app will scale up the number of instances to provide consistent performance.

Billing

We host the instance of our codebase on Google Cloud Platform using App Engine. The project costs around $40 AUD per month. Given that the website is still in infancy, the traffic is very low. If the traffic increases, it would cost more accordingly. Our project currently is enjoying a 365-day free trial with $350 AUD credit. According to the payment charged every month, it's likely that the credit will run out within 5 months. The screenshot below shows the billing details of a month.

Billing detail