perseus.

Case Study

Full Stack Candidate Coding Challenge

Solution Submitted by —
Ronit Mishra

Requirement

Create a dashboard with the provided dataset

Specs:

- A service that exposes the data via API(s) to be consubme by a web page (UI)

- A web page that displays a report with possibility to filter data per user

- Reporting dashboard should display general statistics such as **Average completion time, who completed faster** etc

Auxiliary Stories:

- Any additional reporting, such as **Search by time, Course Completion Rate, Shorter & Longer Course** based on average completion time etc
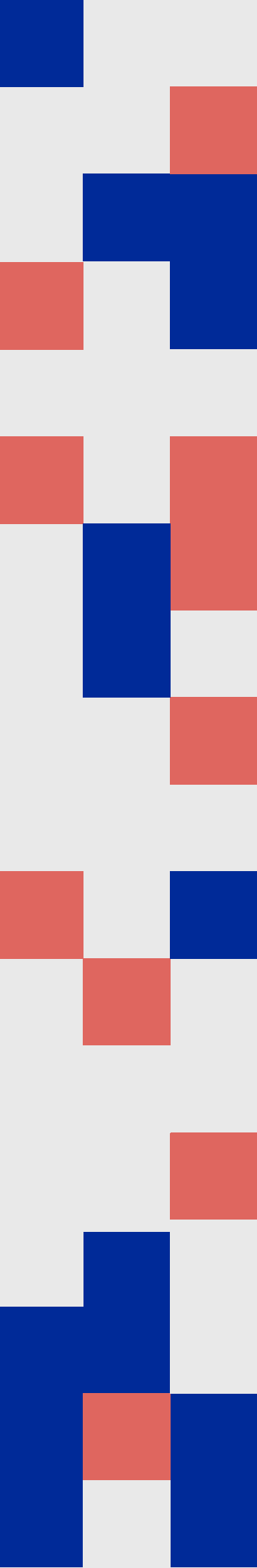
React based Web UI

Dashboard created using **React.js** bootstrapped with **create-react-app**, styled with **Material UI**

A lightweight React.js wrapper around **Chart.js** for visualizations.

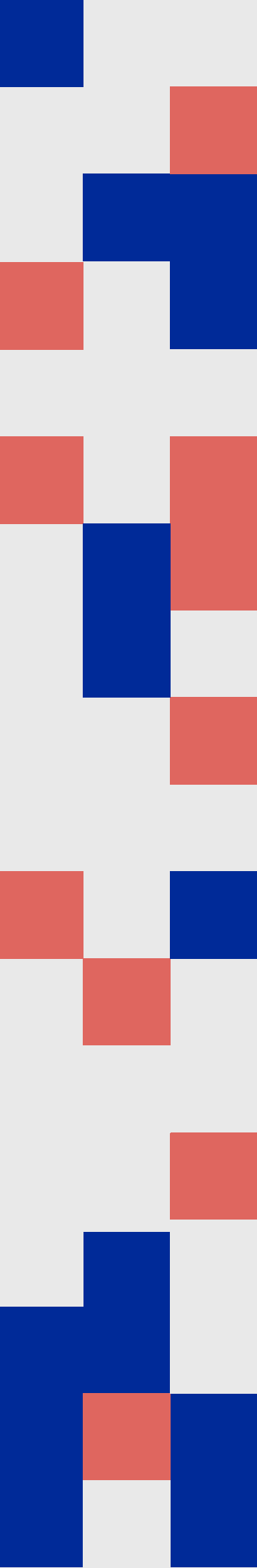For demonstration purpose dashboard will be deployed on **Firebase**

Proposed Solution[1]

RESTful Data Services using Flask

Using **Flask**, a microframework based on **Python** to create **REST APIs** to serve data for the dashboard.
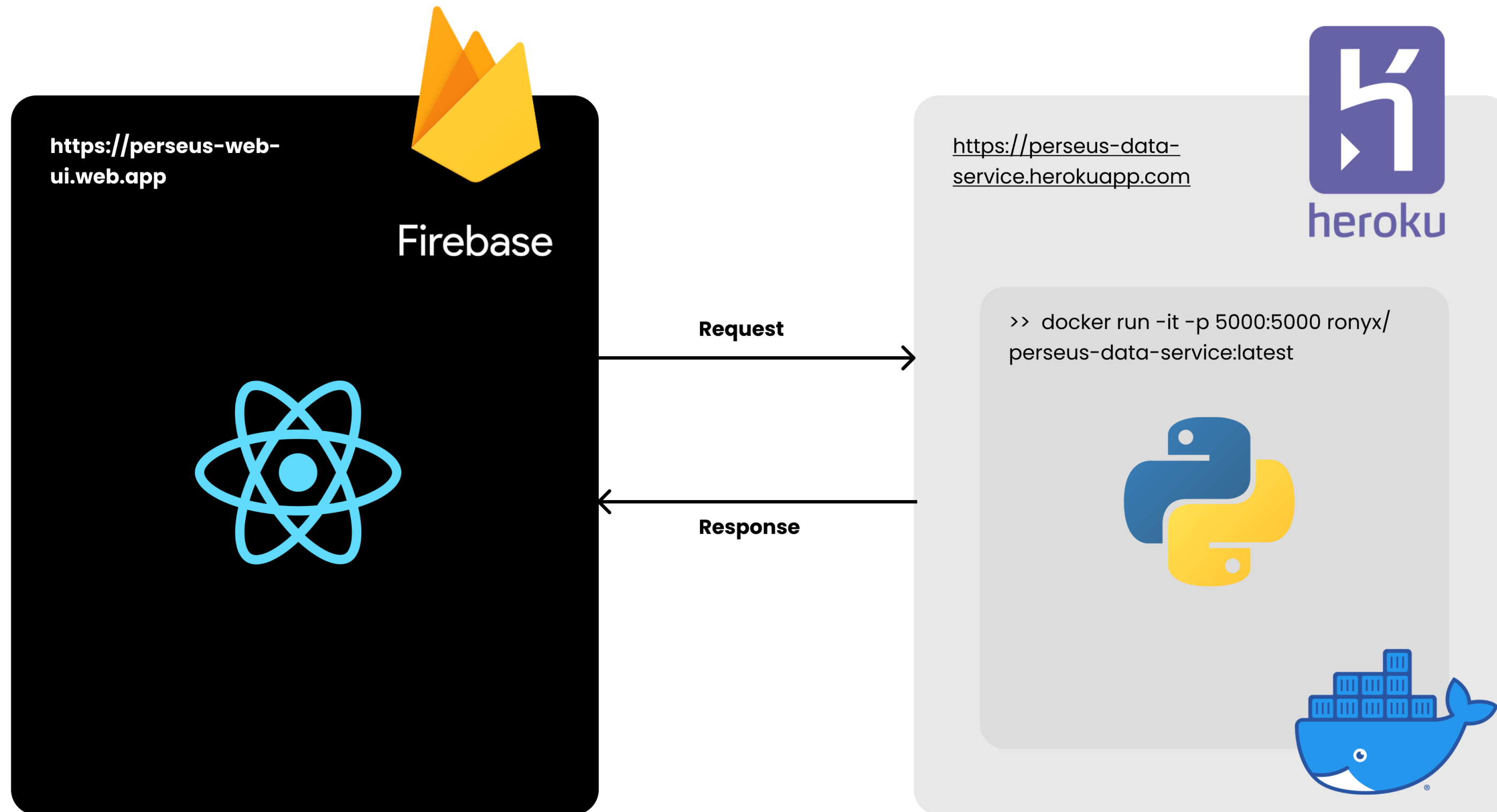
Using **Swagger UI** for to generate interactive documentation based on **OpenAPI Specifications** that lets your users try out the API calls directly in the browser.

For demonstration purpose data service will be **Dockerized** and deployed on a **Heroku** dyno.

Proposed Solution[2]
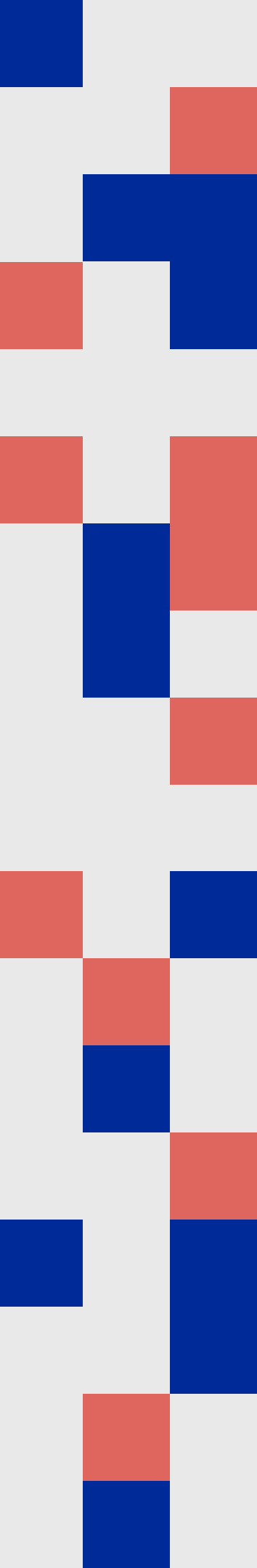
High-level Architecture

https://perseus-web-ui.web.app

Firebase

https://perseus-data-service.herokuapp.com

heroku

>> docker run -it -p 5000:5000 ronyx/perseus-data-service:latest

Request

Response

Deliverables

Case Study Demo:

Dashboard: https://perseus-web-ui.web.app

Data Service: https://perseus-data-service.herokuapp.com

Source code available at the following GitHub repository:

https://github.com/pyronlaboratory/perseus

References

Following online resources were used during the development of this case study

https://reactjs.org/

https://material-ui.com/guides/api/

https://flask.palletsprojects.com/en/2.0.x/

https://devcenter.heroku.com/

https://firebase.google.com/docs

https://docs.docker.com/

and good old.. https://stackoverflow.com/