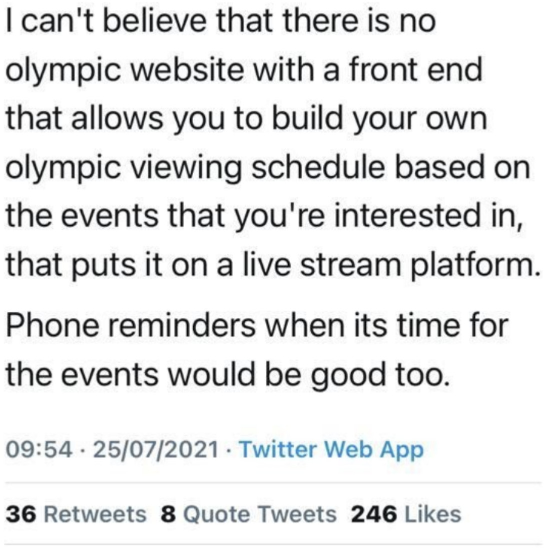
Group Members: Melanie, Faiqa, Ellie, Meg

What are you building?

An app that will allow an user to input their favourite Olympic sports, and the app will return a personalised Olympic schedule. We have called this app “Olympick” (as you ‘pick’ your Olympics).

What does it do or what kind of problem does it solve?

Currently there is no way to create a viewing schedule for the Olympics that only includes specific sports.
We are solving a problem with the Olympic viewing experience.
We were inspired by this tweet:



I can't believe that there is no olympic website with a front end that allows you to build your own olympic viewing schedule based on the events that you're interested in, that puts it on a live stream platform.

Phone reminders when its time for the events would be good too.

09:54 · 25/07/2021 · Twitter Web App

36 Retweets 8 Quote Tweets 246 Likes

[1]

What are the key features of your system?

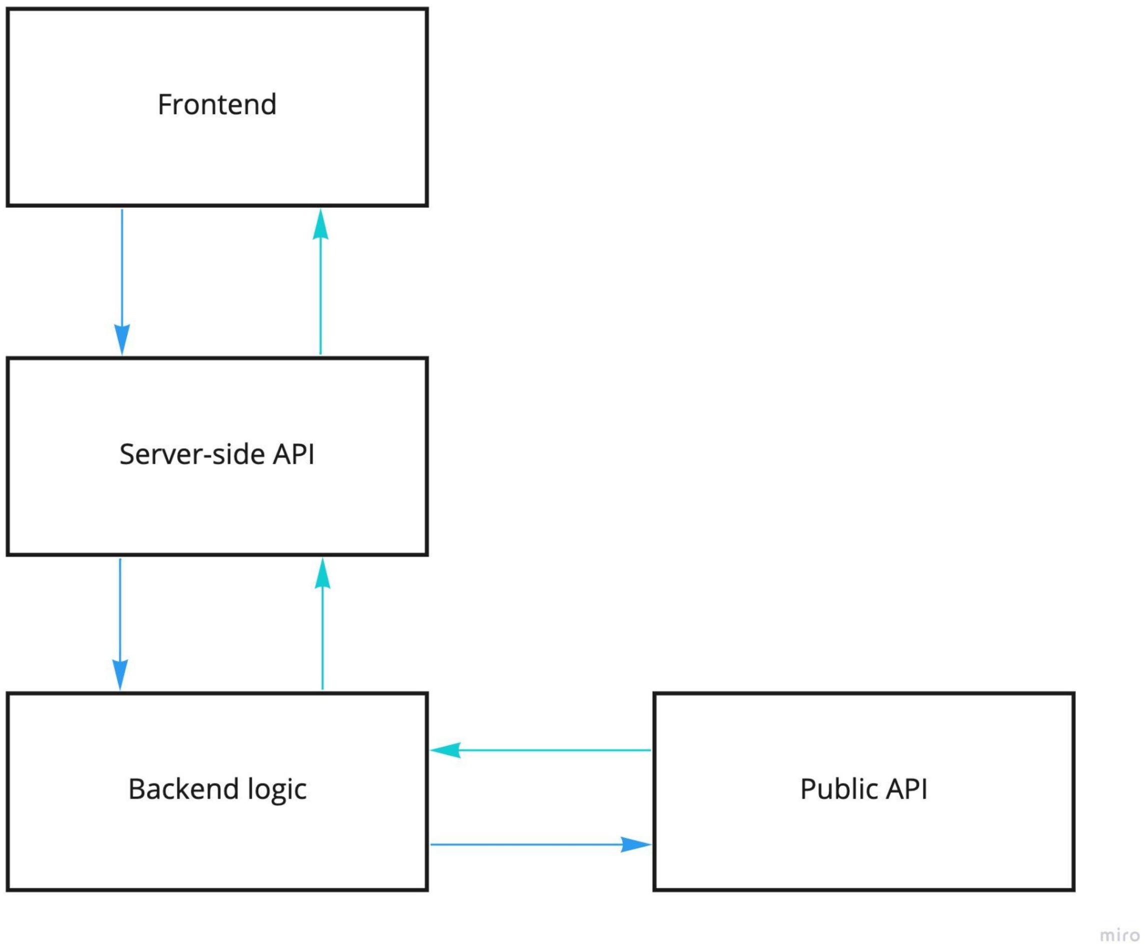
- Front-end
  - Search bar/form/drop-down menu for the input of sport name
  - Filtered results (schedule) presented
  - Footer and navbar
  - Extended goal: Log-in and authentication
  - Extended goal: An email reminder some time before an event
  - Extended goal: Able to sync events schedule into Google calendar
  - Extended goal: Design the front-end like a calendar
  - Extended goal: Automated program execution
- Public API
- Backend logic
  - Flask used to create server-side API
  - GET request used to call the public API (or POST)
- Test files (Python)

We will be using HTML/CSS or React for our front-end, and Python for our back-end.

[1] (Twitter user’s identity removed for privacy protection).

Group Members: Melanie, Faiqa, Ellie, Meg

Provide a sample architecture diagram of your system



Describe the team approach to the project work: how are you planning to distribute the workload, how are you managing your code, how are you planning to test your system.

Team approach:
We will be highlighting our skills and strengths, and helping each other to improve in our weaker areas by coding interactively (e.g. watching each other code and discussing this). We aim to be as collaborative and communicative as possible, which will improve both our technical and 'soft' skills. All members are equally responsible for the quality of the work and completing their work within a given timeframe.

Workload distribution:
We have created a Trello board (of project requirements) and a SCRUM board (for project progress) that will help us to understand what work needs to be distributed. We will be having a daily SCRUM meeting so that work can be distributed flexibly - e.g. if someone is taking a little longer to do a portion, someone else can help or that person will not be assigned any more work. We will be able to communicate

effectively about our individual progress this way, and help each other effectively also. We will be using Agile methodologies moving forward.

We conducted a SWOT analysis to portion the work according to people's preferences and strengths. We found that we have very similar skill sets (e.g. back-end programming, OOP principles) but also individual advantegous experiences (e.g. regular SQL coding, project management). We have decided that we will portion the work equally between members and between the front and back-end, as we have similar strengths and all wish to improve in other areas.

Code management:

We are managing our code using Git and GitHub.

We are going to create a repository for our code, and separate branches for different features which we will eventually merge. We will be reviewing each other's pull requests and we have elected Ellie to manage our repository.

Testing:

We are going to use the unittest module in Python for unit testing, and will mock all external inputs (the user input, the public API etc).