Software Engineering Immersive Course: Project 3 - MERN Stack App

techTalk

Brief

Build a full stack app by making your own front and back end using MERN (MongoDB, Express, React.js & Node.js). The app will most likely have multiple relationships and CRUD functionality for at least a couple of models.

Timeframe: 1 week, group coding collaborating with Vikram Bageja, Reema Patel and Tom Good

Summary

Tech Talk is full stack app for arranging and searching for tech related events. Users can create events, see who is attending and comment on events that they are attending. We built the whole app in JavaScript. Our back-end used an Express API to serve the data from a Mongo database and is consumed by our front end built in React.

This was a group project over a period of a week, that required rigorous planning and constant collaboration. We used a combination of Visual Studio Live Share to group code and pair programming to set up the framework, then we split to work on individual elements. Git and Trello were instrumental here for version control and ensuring our code wasn't duplicated or conflicting. I focused on the forms - styling these with Skeleton, the language buttons - using Adobe Illustrator and Photoshop, and the map element using Mapbox API and React-map-gl.

Deployment

Access techTalk here on Heroku.

Installation Instructions

To run the app from the source code, use the clone button to download the source code from GitHub. From the root directory type "yarn","mongod", "yarn seed" and "yarn start" in the terminal. The project will run on localhost:4000, and will be viewable on any web browser.

Technologies & Methodologies Used

- HTML 5
- CSS 3
- JavaScript (ES6)
- MongoDB (NoSQL)
- Mongoose
- Express.js
- Node.js
- React.js
- React Router
- Skeleton
- SASS
- Moment.js
- Mocha
- Chai
- Supertest
- Webpack
- Yarn
- Insomnia
- Git
- GitHub
- Heroku
- Adobe Illustrator
- Adobe Photoshop
- Wireframes
- MVC Architecture
- Authentication
- RESTful APIs
- CRUD Functionality
- CSS Transitions

Features

- Users are able to:
  - register, login and out
  - filter events by location, programming language and cost
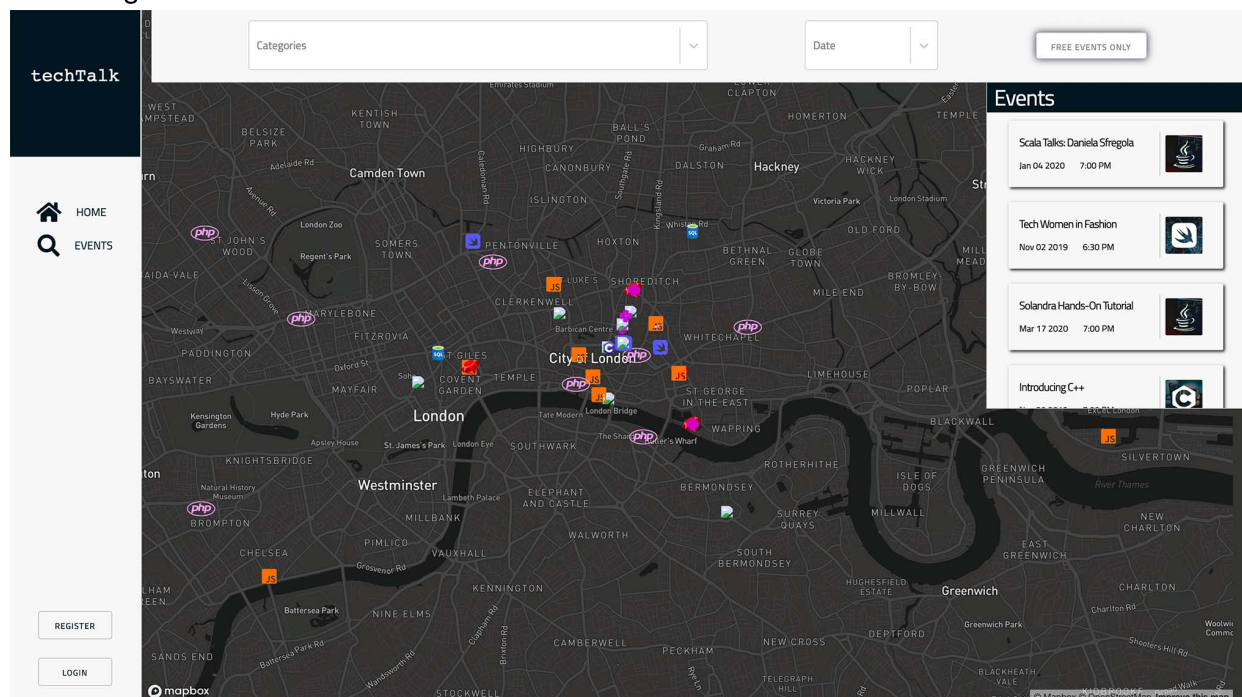
- sign up to attend events
  - create, edit and delete their own events
  - comment on each others events
- rollover programming buttons
- Mapbox search functionality
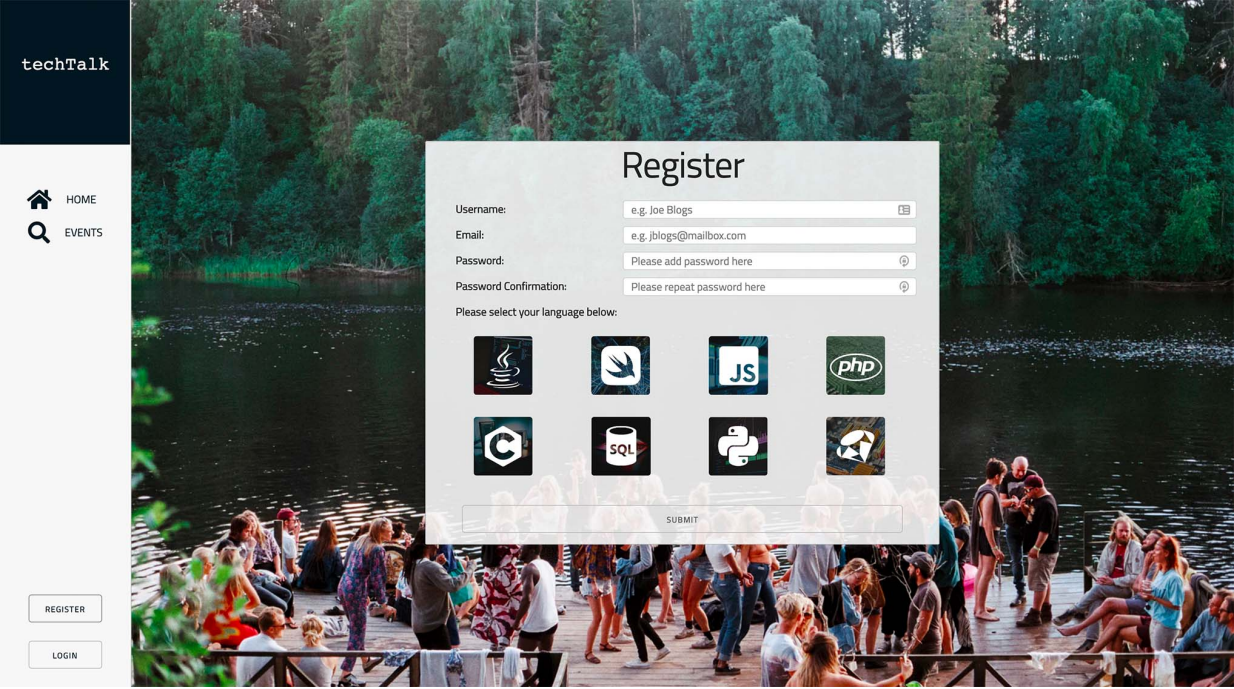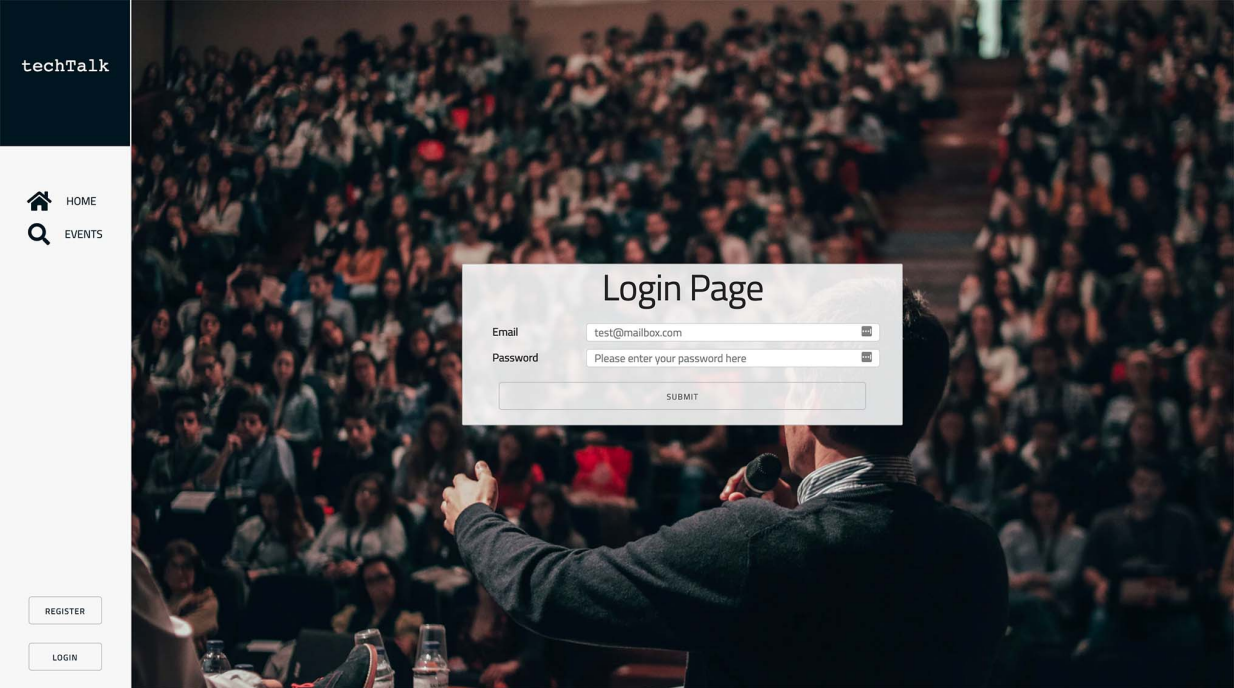- Inline text editing

Website Architecture

Home Page

Event Page

Register Page

Login Page

Profile Page

Event Create Page

Event Page - editable by the host

Filtered Events Page

Takeaways

It was the first time we had linked a front and back end so this threw up a handful of challenges. Although communication broadly speaking was very good, going forward I will be sure to spend more time pseudo coding in advance to ensure detailed clarity on all elements.

Importance of:

- Good communication:

  - commenting code to ensure everyone was able to see how elements they hadn't written worked

  - Regular git commits with helpful comments

- maintaining clarity on which features are core to the MVP, and which you can work on at a later date

- writing automated tests for our RESTful resources on the back-end, save time - particularly as the apps become more complicated

---

Challenges Overcome

- programming language search bar functionality
- using Mapbox
- avoiding duplicating code and causing conflicts
- agreeing on styling preferences for uniformity across the site

---

Future Improvements

- fully responsive design
- develop a booking/payment system for events
- direct chat facility
- add more filters to search by
- increased functionality in the profile settings
- event image showing on show page

---

sophieturnell.com
linkedin.com/in/sophieturnell/