Project Report

Group 2:

David Babcock

* ERD
* Project Report
* Requests List Page and Accepting/Declining logic
* Super Admin Interface
* Styling (Bonus)

Patrick Bauer

* Authorization
* Home Screen
* Password Hashing (Bonus)
* Reset Password (Bonus)
* Purchased Domain (Bonus)
* Setting up Google Cloud and Server (Bonus)

Samuel Arminana

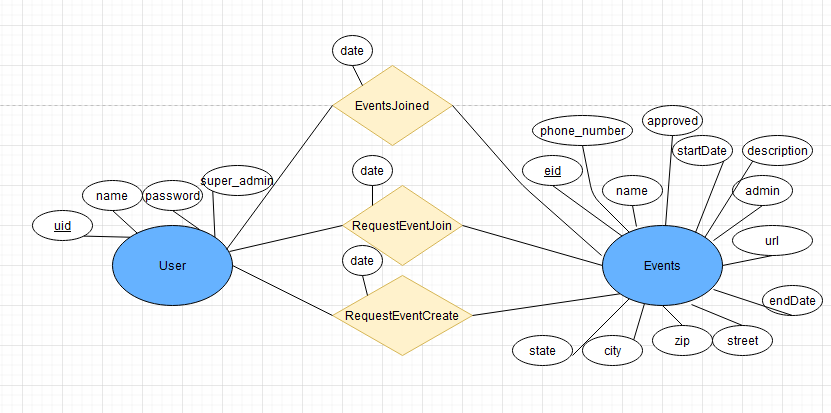
* Admin Interface
* Create an Event Form
* Bug Fixing
* Styling (Bonus)

**Contribution**: I implemented the Admin Interface and Create event form along with the accompanying confirmation screen which features a button to return to the Admin Interface. I also did general bug fixing such as, hiding a navbar option if your access is limited, and hiding the sign out and reset password buttons if you are logged out. I also implemented styling into some screens like the Participant Interface.

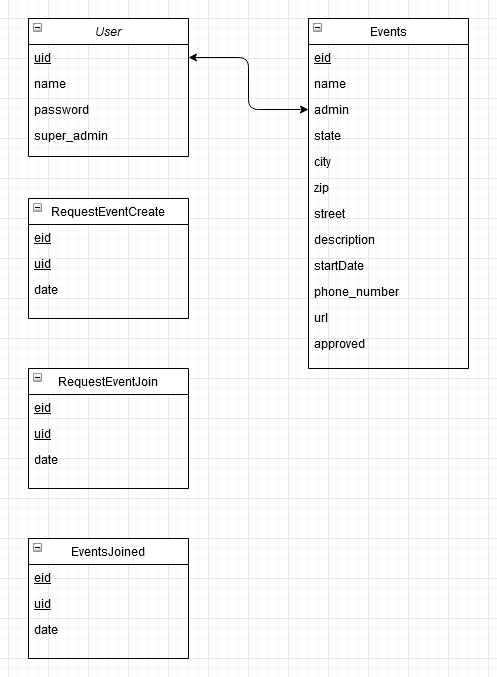
Maya Awad

* Participant Interface
* Events Searching Interface
* Styling (Bonus)

I: Database Design



We decided, for simplicity, that there should be two entities, Events and Users. The user is defined by a uid, name, password, and super\_admin. Events are defined by eid, name, location fields, the admin (a user uid), start and end date, phone\_number, website url, and if it has been approved. The three additional tables are EventsJoined, which shows what events a user has RSVP’d, RequestEventJoin, which shows which user has requested to join an event, and RequestEventCreate, which shows the events awaiting approval. In hindsight, there is some redundancy in our tables because the approved field can stand in for the EventsJoined table.



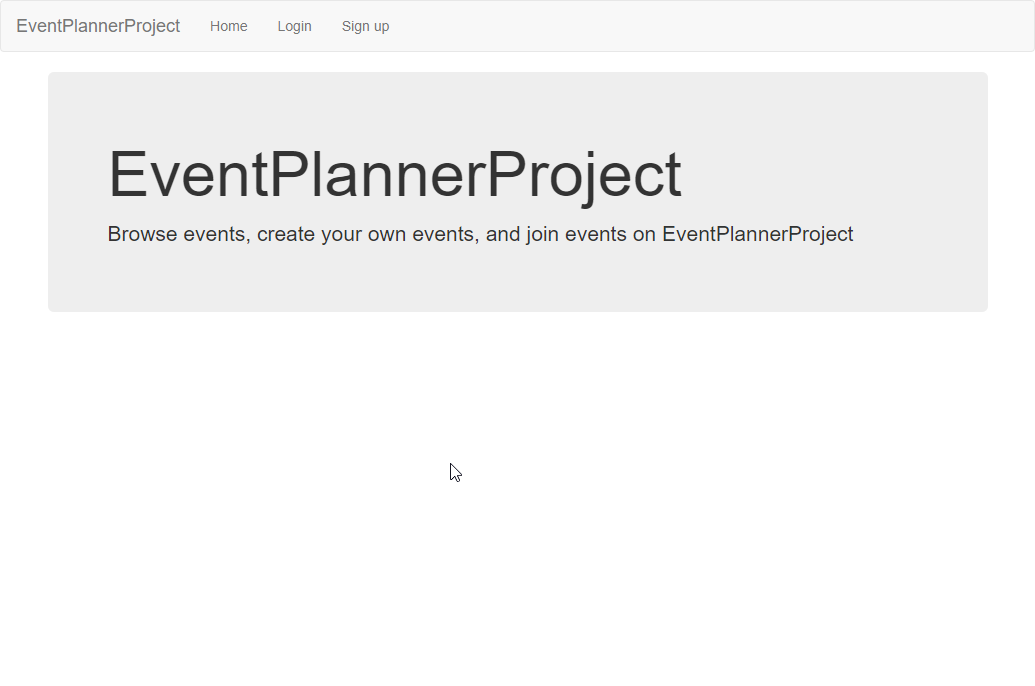
II: Software

To keep our technology stack in line with the class, we decided to go with a Linux, Apache, PHP, and MySQL stack. Few of us had experience with PHP, but with a couple quick tutorials, we were able to get up to speed. PHP allows html to be embedded in its files, so it helped to keep the number of files down. To debug, we used WAMPserver to set up a local development environment which has a self-contained Apache, PHP, and MySQL in it. This allowed us to work with a database offline and test our code. To style our webpages, we used bootstrap to make our pages look presentable.

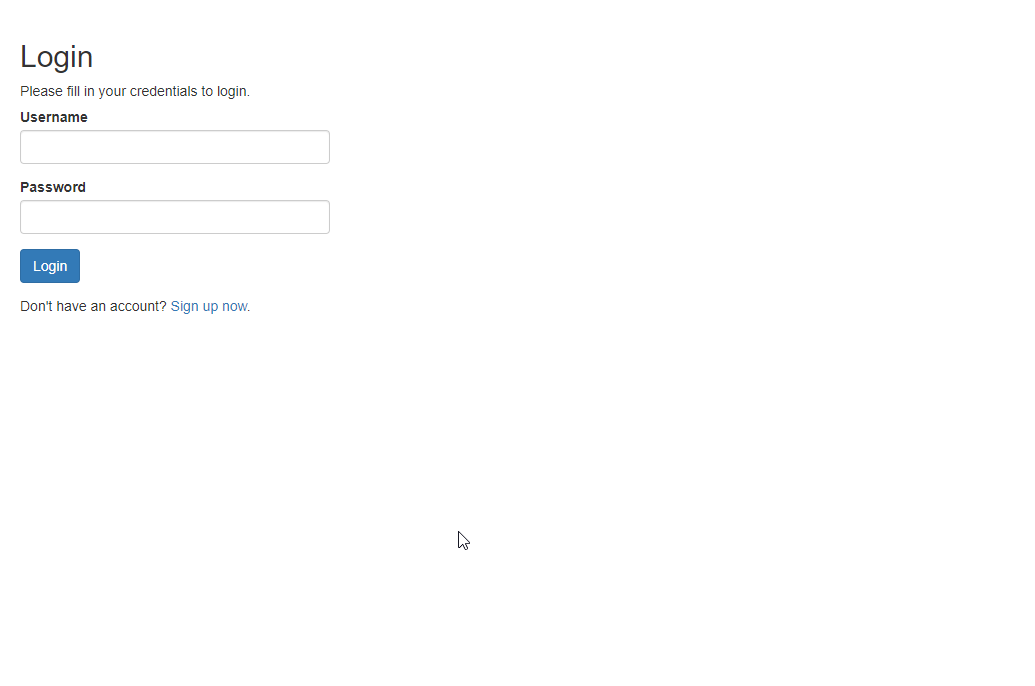
We then deployed to a Google Cloud Server and linked it to a purchased domain and ran the website through that for presentation and final development.

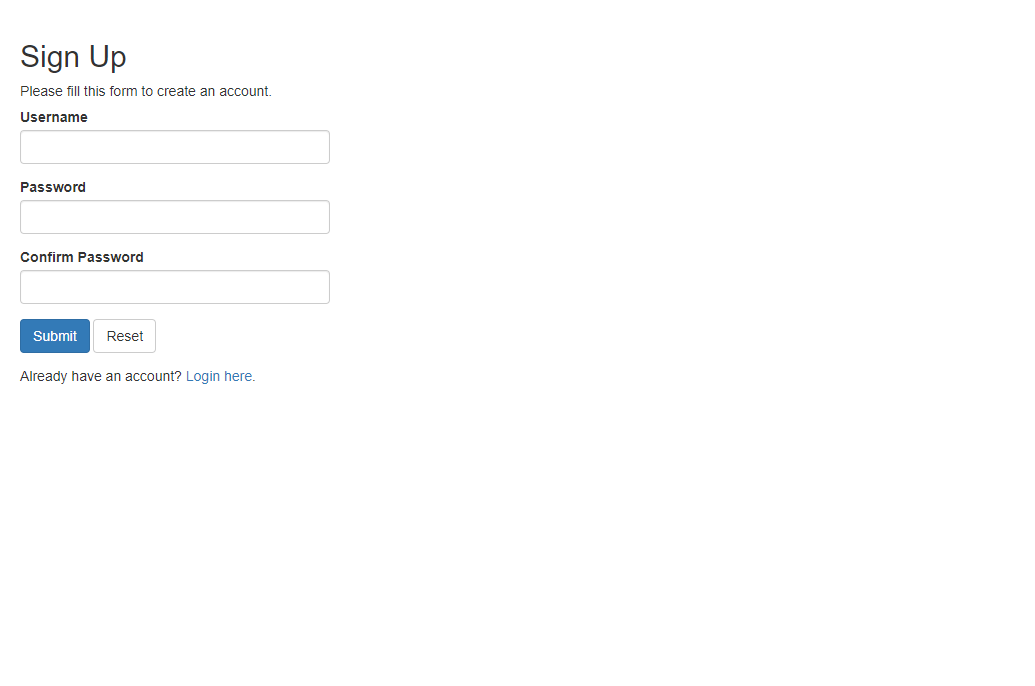
To connect to the database, the tutorial I (David Babcock) followed used mysqli\_connect, which creates a connection to the database with the database credentials. A query can then be made from this connection and each row can be accessed through the row function in a while loop. Each individual column can be accessed if the name of the column is known. Then, the html can be printed to the page with the values returned from each row.

Welcome Page:

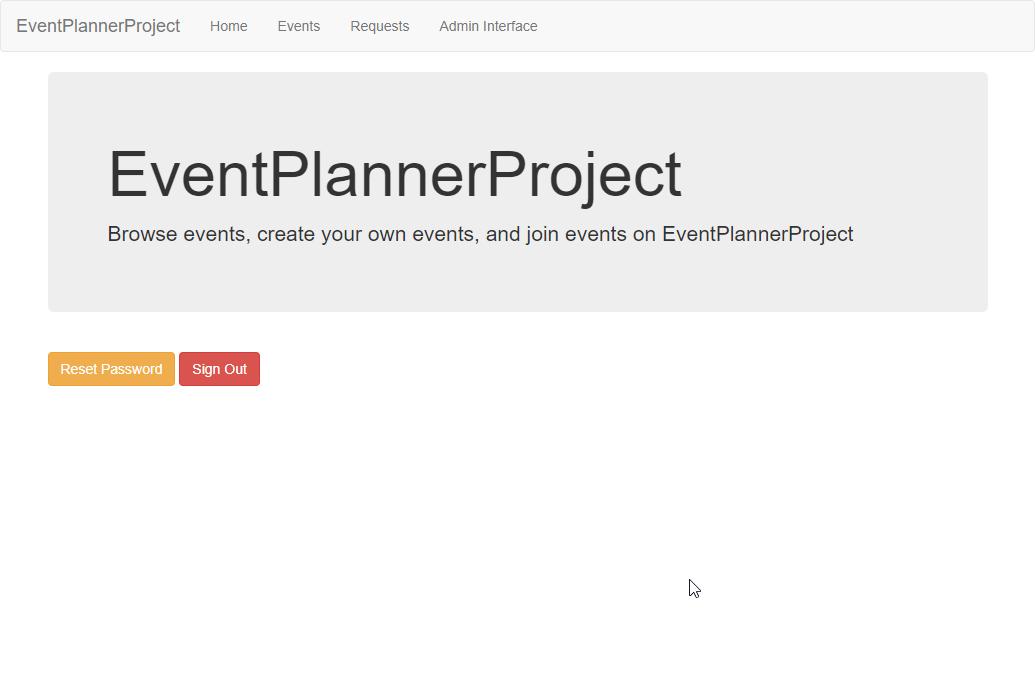


Login Page:

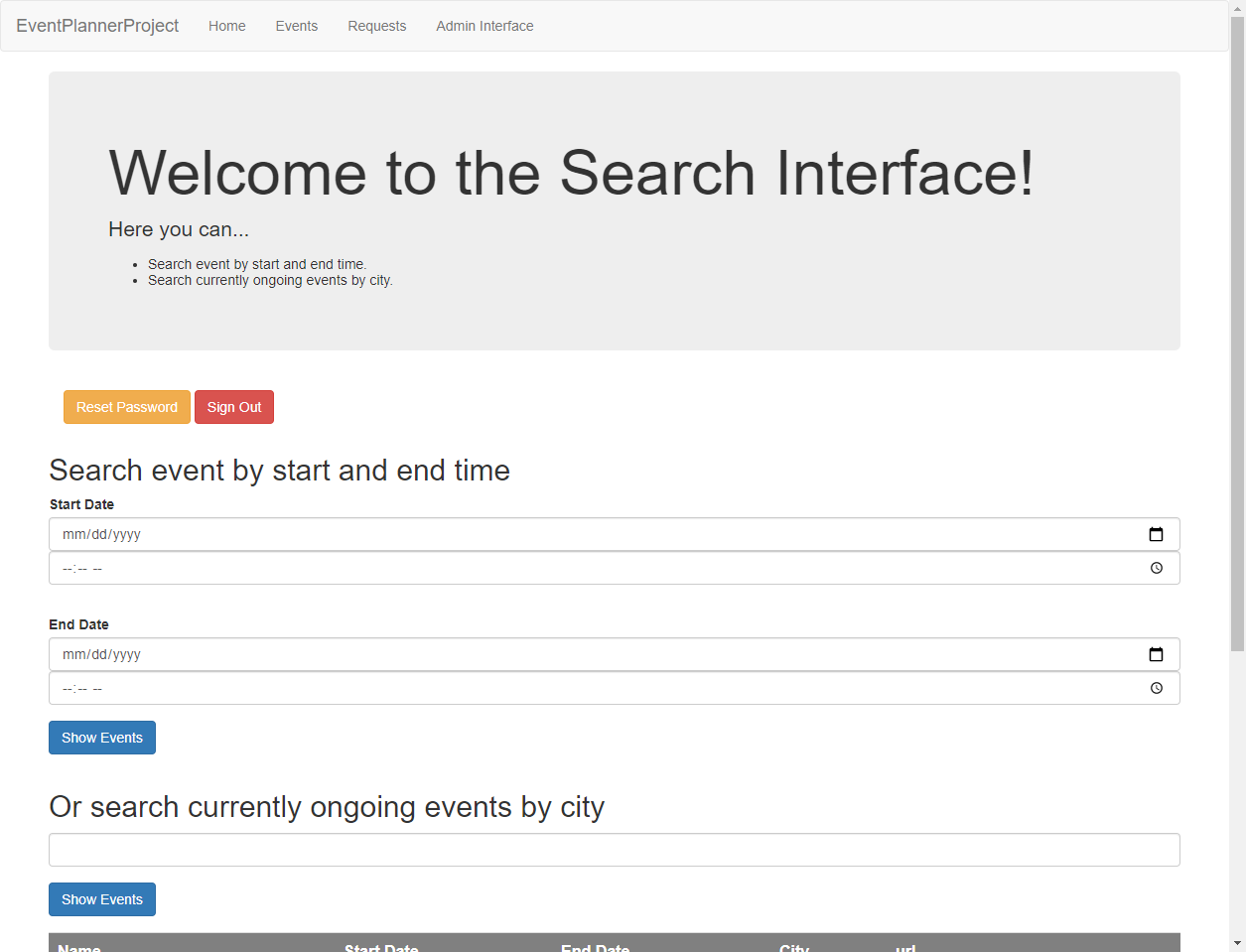


Register Page: 

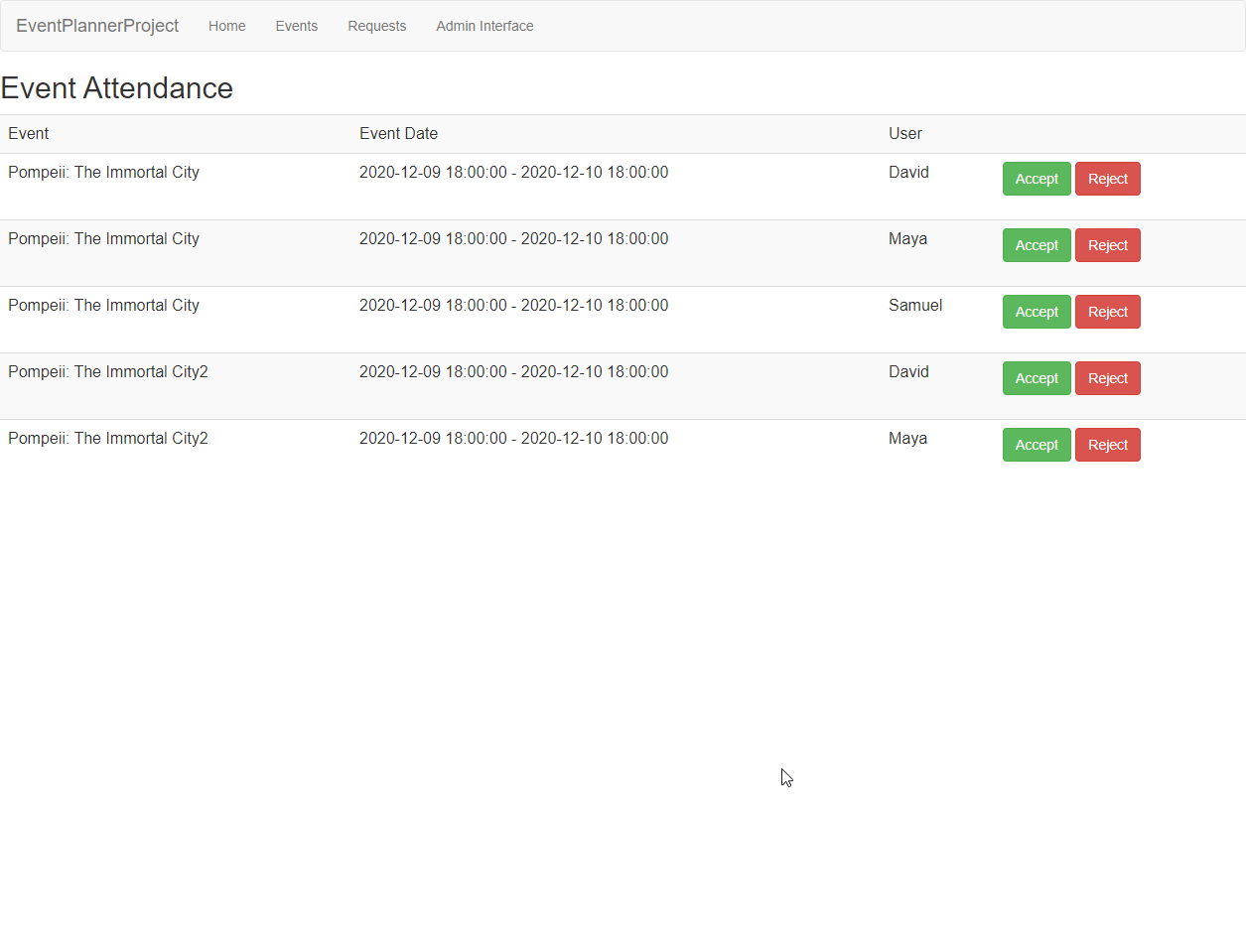
Home Page:



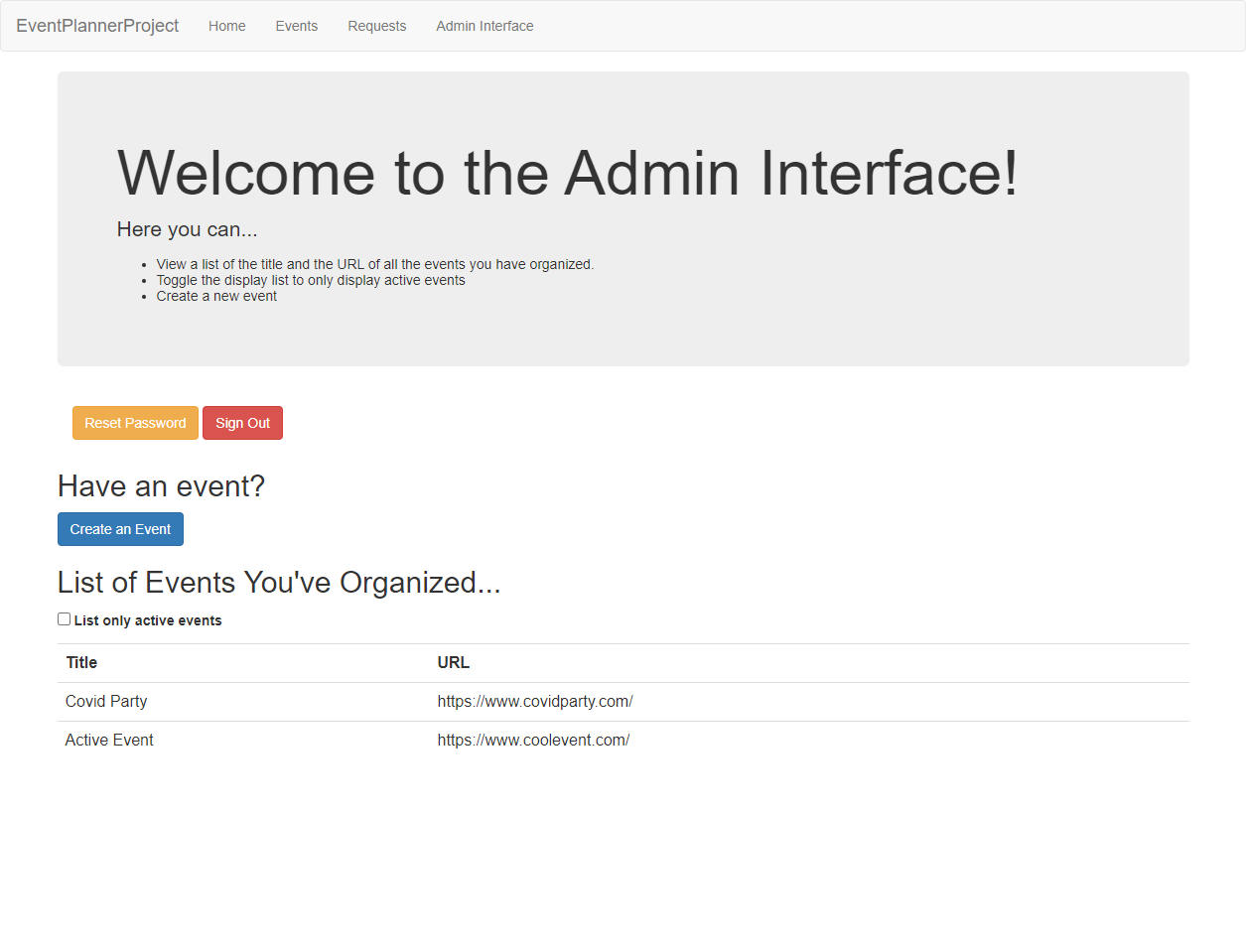
Events Page:



Requests Page:



Admin Interface Page:



Super Admin Page:

