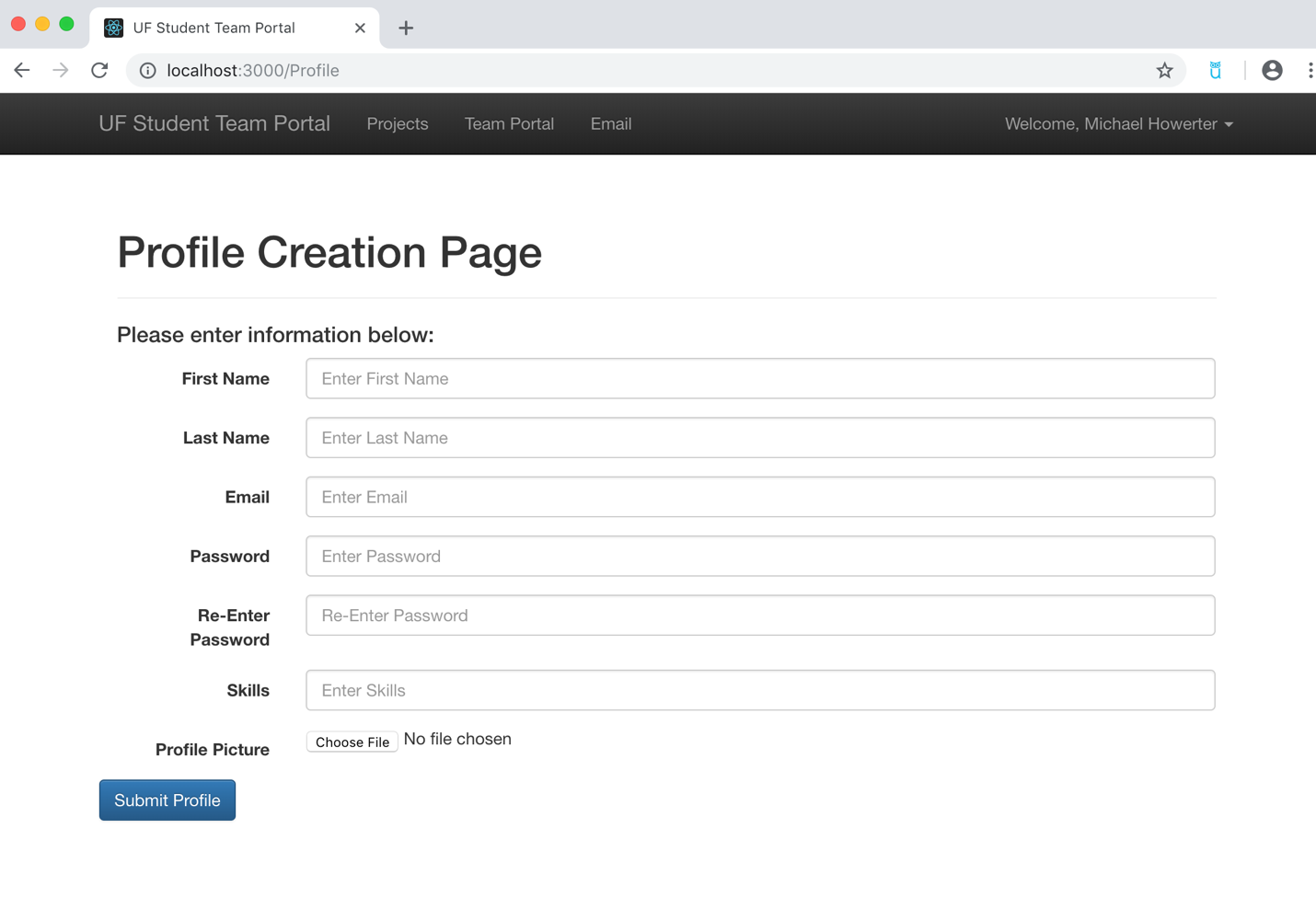
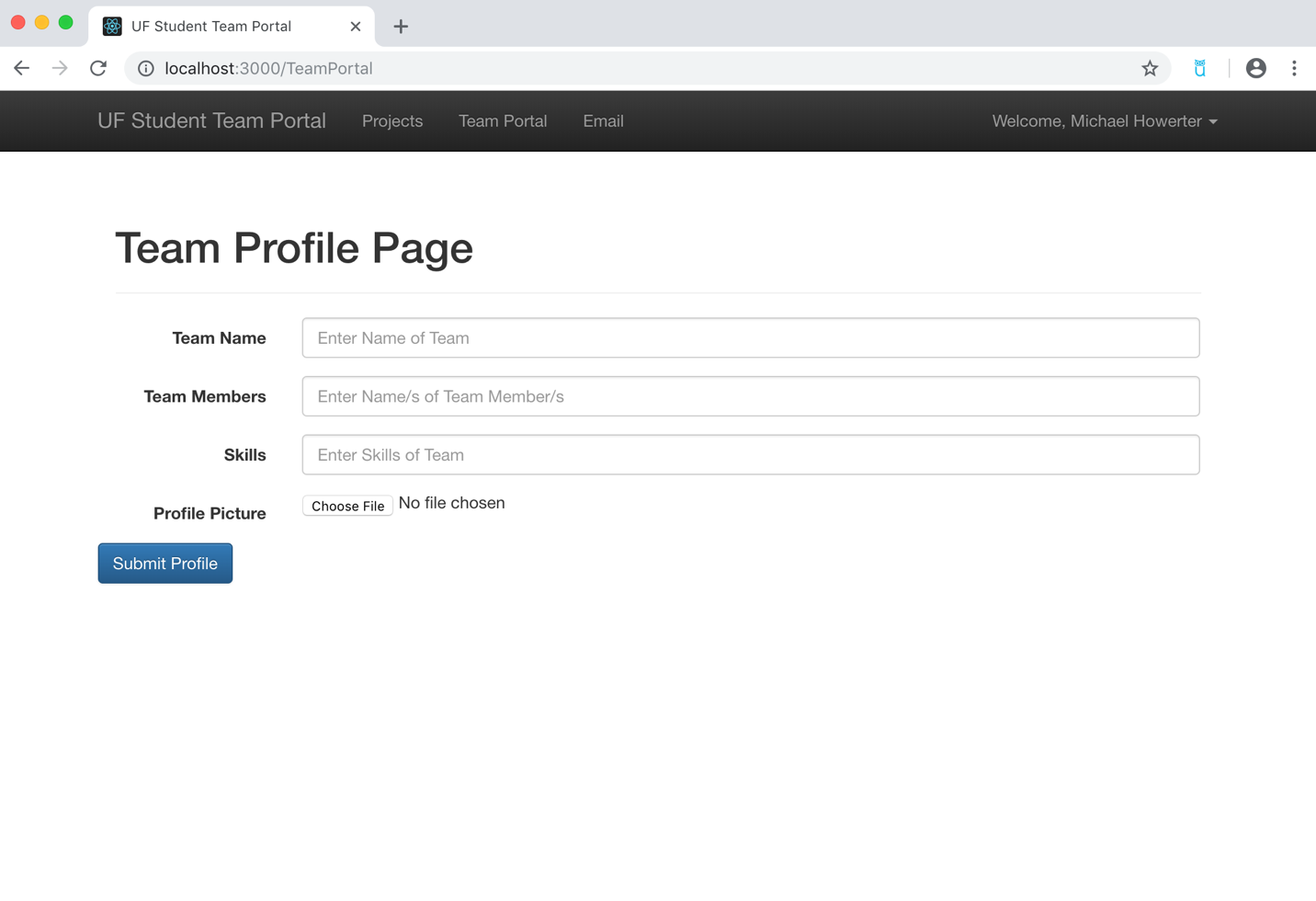
**Features implemented:**

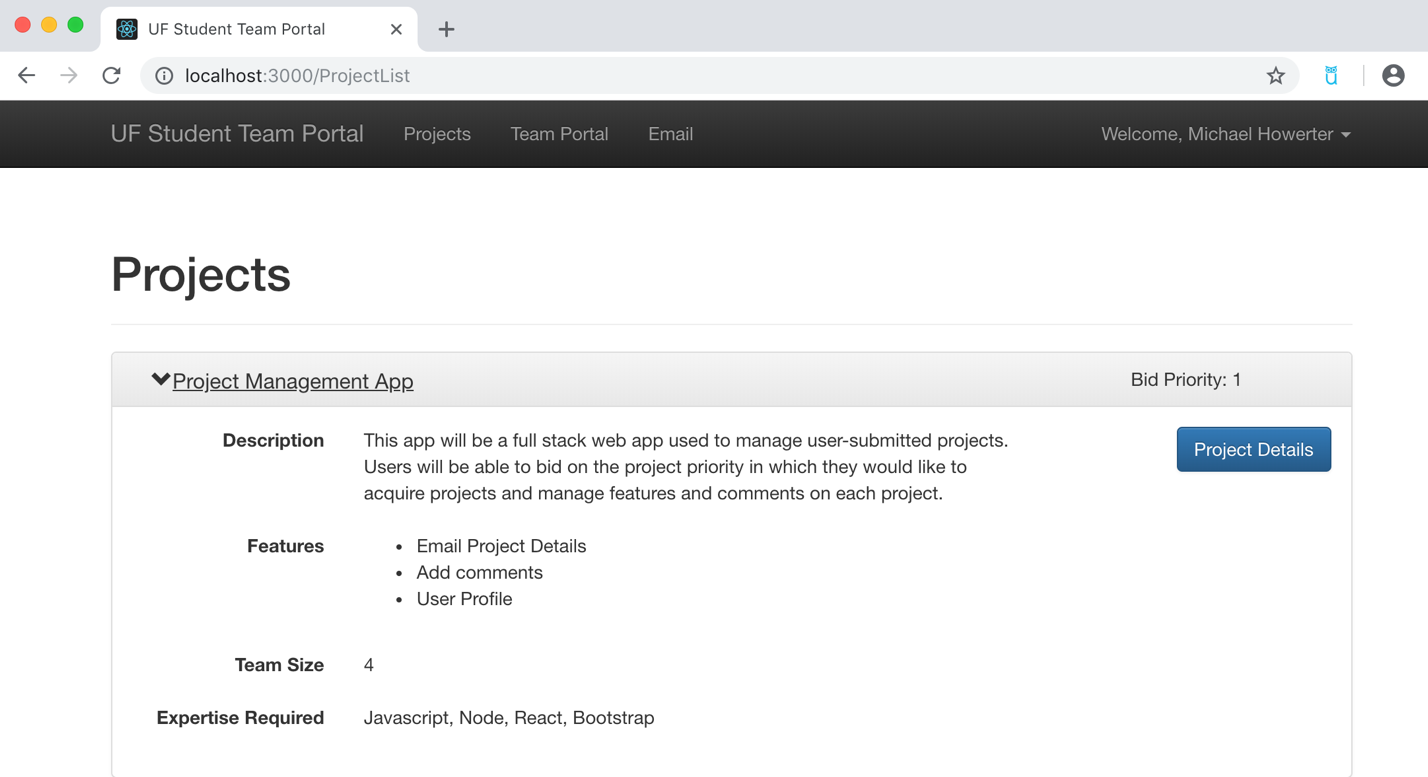
Individual Profiles



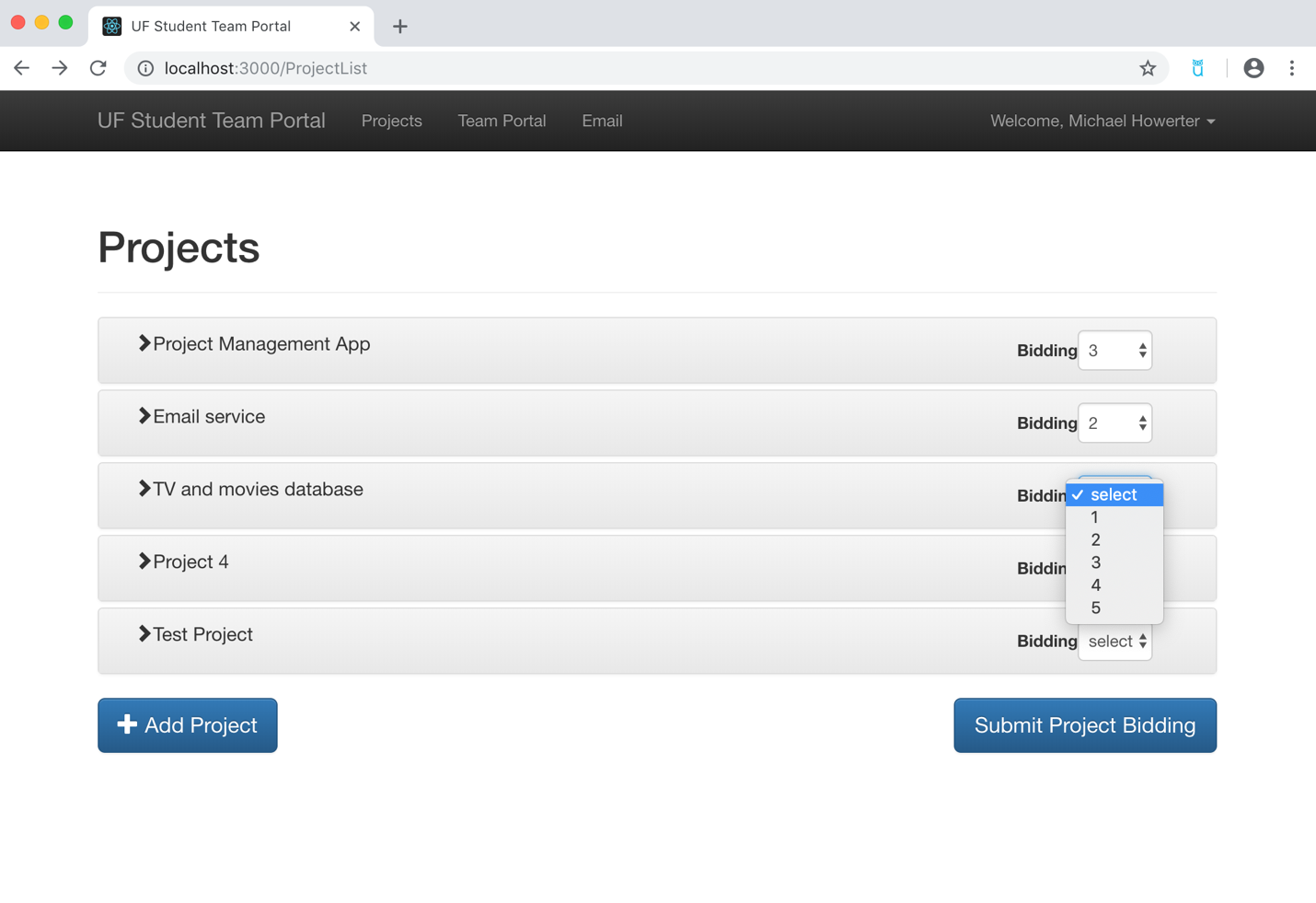
Team Profiles



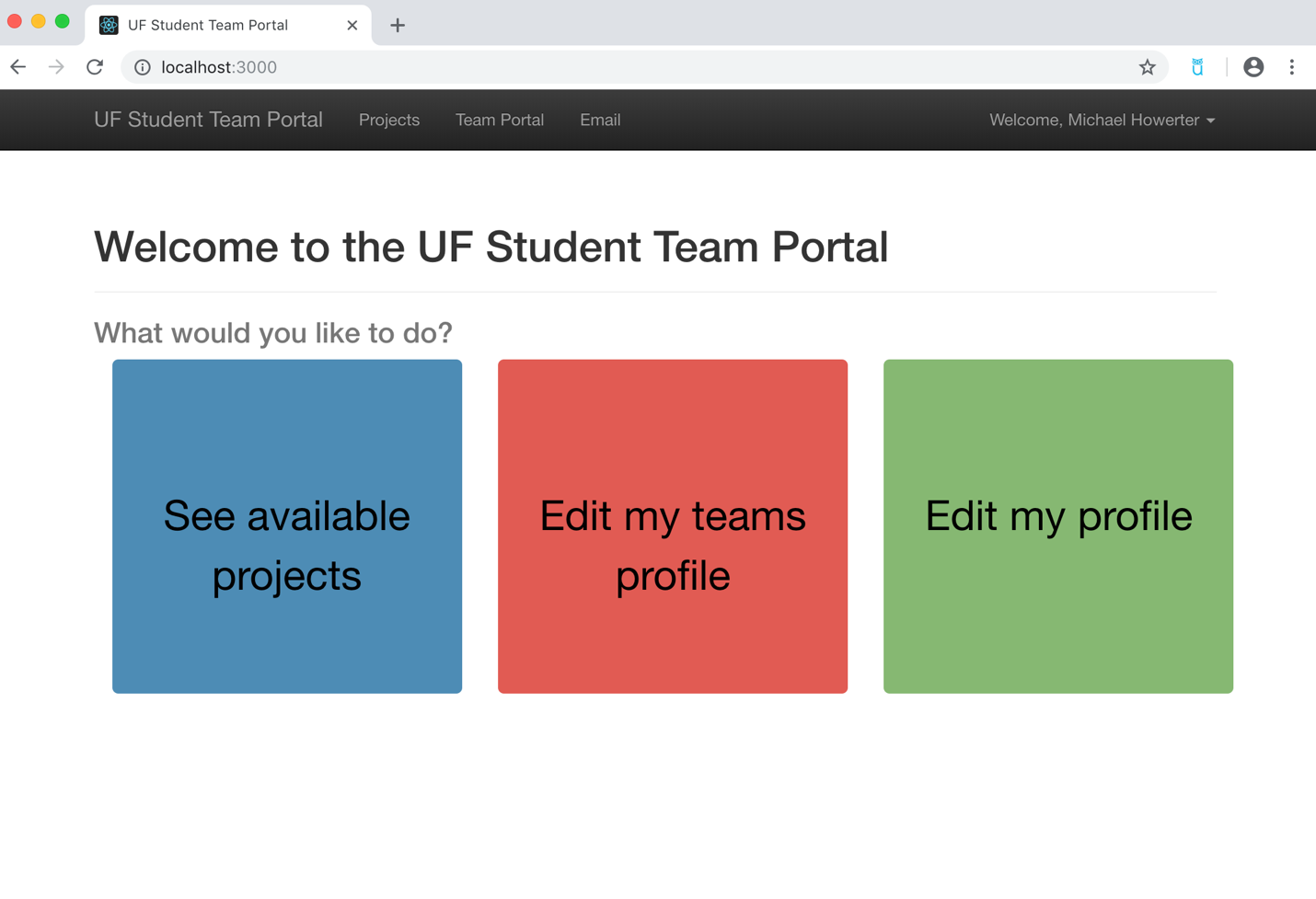
Project Details



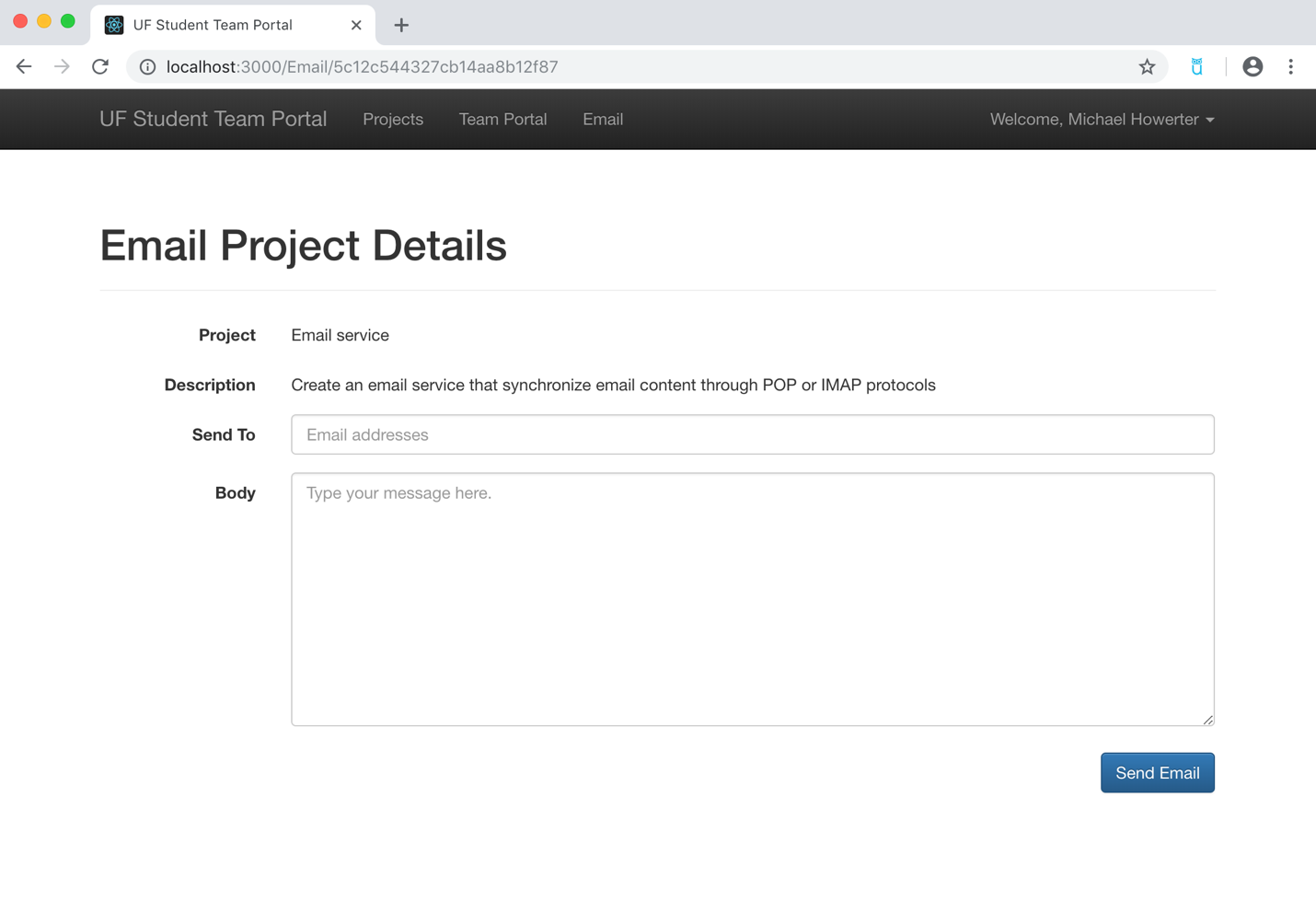
Project Bidding



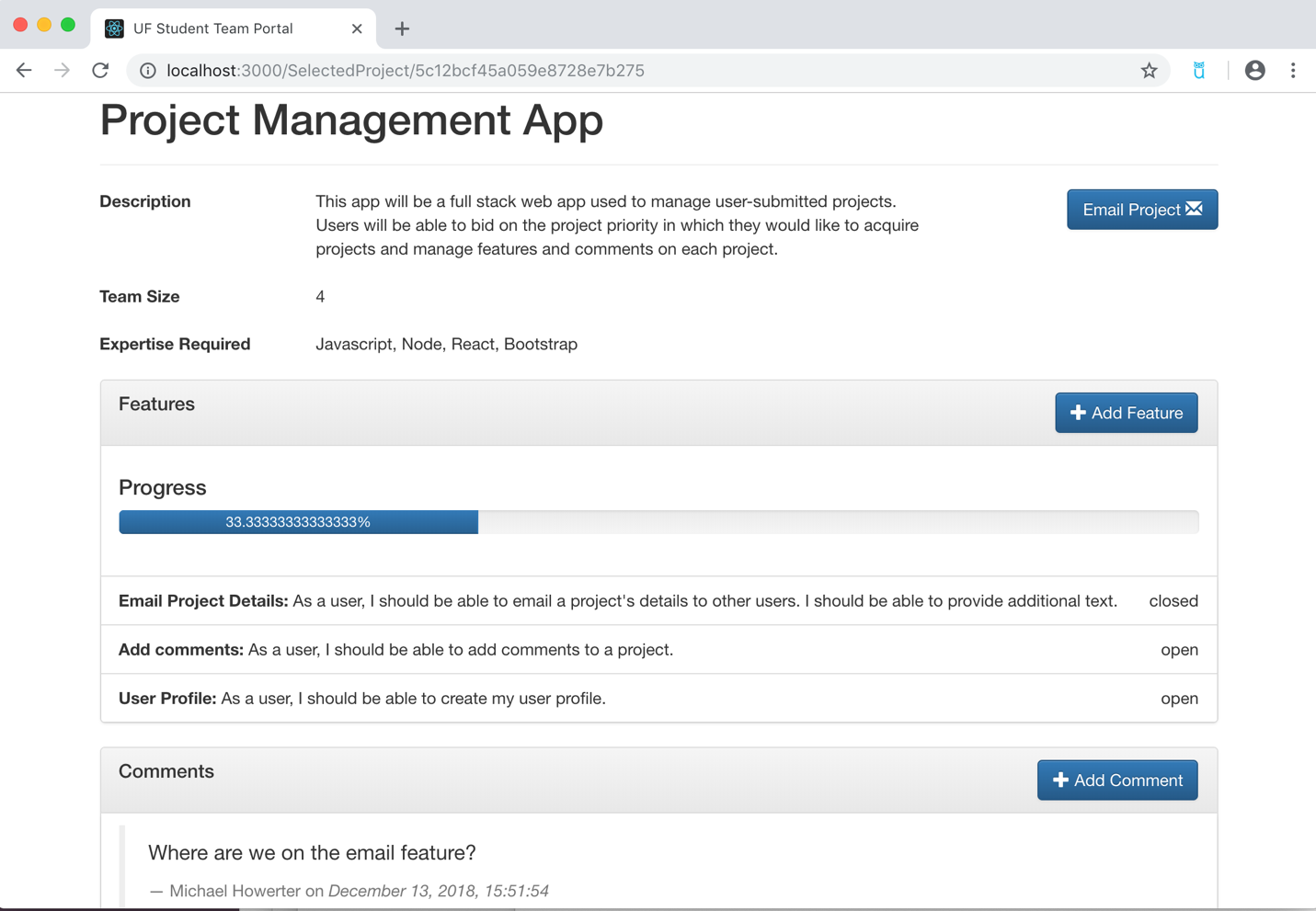
Home Page



Email Page



Project Status/Updates



**How to run the app locally:**

1. npm run installDeps

2. npm start

- It should open the browser at http://localhost:3000/ with the initial Application

**How to update database and server connections**

Database is configured in server.js.

We are using mLab to store our database.

New developers should create a new account in that service ( https://mlab.com/signup/) and continue from there

# REST endpoints :

# Projects

## Create

URL :

- PUT http://localhost:90/api/projects

Example Request :

{Questions for you guys:

3. Did any of you keep track of an individual burndown chart? If not, I’m going to estimate based off of GutHub commits.

Things you should know:

3. I submitted a random version of the wireframes because I couldn’t tell which ones were the most recent and I don’t think it matters.

4. Part V asks about automated testing. We didn’t do any so I’m not submitting anything for this section.

"name": "Project3",

"description": "Project 4 Example"

}

## List

URL :

- GET http://localhost:90/api/projects

Example Response :

[

{

"\_id": "5bf3959b1e53214ea0b80613",

"name": "Project1",

"description": "Project 1 Example",

"\_\_v": 0

},

{

"\_id": "5bf395c21e53214ea0b80614",

"name": "Project2",

"description": "Project 2 Example",

"\_\_v": 0

},

{

"\_id": "5bf395c91e53214ea0b80615",

"name": "Project3",

"description": "Project 3 Example",

"\_\_v": 0

}

]

## Get Project by ID

URL :

- GET http://localhost:90/api/projects/{id}

Example Response :

{

"\_id": "5bf3959b1e53214ea0b80613",

"name": "Project1",

"description": "Project 1 Example",

"\_\_v": 0

}

## Delete project

URL :

- DELETE http://localhost:90/api/projects

Example Request :

{

"id": "5bf395cf1e53214ea0b80616"

}

# Users

Unique identifier: email

## Create

- URL :

- PUT http://localhost:90/api/users

Example Request :

{

"firstName": "Peter",

"lastName": "Parker",

"email" : "peter@hotmail.com",

"password" : "hello123",

"photo" : "https://i.imgur.com/rzPz4qG.jpg",

"skills" : "javascript,java,html,nodejs"

}

## List

- URL :

- GET http://localhost:90/api/users

Example Response :

[

{

"\_id": "5bfe26b1b509011b709f6501",

"firstName": "Peter",

"lastName": "helloWorld",

"email": "peter@hotmail.com",

"password": "EdTdw1fggilo2/0ia24cKqwBjQdqVNpPZeHcgYBoSsM=",

"photo": "https://i.imgur.com/rzPz4qG.jpg",

"skills": "javascript,java,html,go",

"\_\_v": 0

}

]

Get User by ID

- URL :

- GET http://localhost:90/api/users/{id}

Example Response :

{

"\_id": "5bfe26b1b509011b709f6501",

"firstName": "Peter",

"lastName": "helloWorld",

"email": "peter@hotmail.com",

"password": "EdTdw1fggilo2/0ia24cKqwBjQdqVNpPZeHcgYBoSsM=",

"photo": "https://i.imgur.com/rzPz4qG.jpg",

"skills": "javascript,java,html,go",

"\_\_v": 0,

"teamID": "5bfe2be542c48127fc5d5e6f"

}

## Update

- URL :

- POST http://localhost:90/api/users

Example :

{

"firstName": "Peter",

"lastName": "Parker",

"email" : "peter@hotmail.com",

"password" : "helloWorld",

"photo" : "https://i.imgur.com/rzPz4qG.jpg",

"skills" : "javascript,java,html,go",

"teamID": "5bfe2be542c48127fc5d5e6f"

}

## Delete

- URL :

- DELETE http://localhost:90/api/users/{id}

Example :

{

"id": "5bfe26b1b509011b709f6501"

}

# Teams

Unique key : name

## Create

- URL :

- PUT http://localhost:90/api/users/{id}

Example Request :

{

"name": "Team A",

"skills" : "javascript,java,html,nodejs",

"members" : "member1,member2,member3"

}

## List

- URL :

- GET http://localhost:90/api/teams

Example Response :

[

{

"\_id": "5bfe2be542c48127fc5d5e6f",

"name": "Team A",

"skills": "javascript,java,html,nodejs",

"members": "member1,member2,member3",

"\_\_v": 0

}

]

## Get Team by ID

URL :

- GET http://localhost:90/api/teams/{id}

Example Response :

{

"\_id": "5bff676f700139209c52e93e",

"name": "Team B",

"skills": "javascript,java,html,nodejs",

"members": "member1,member2,member3",

"\_\_v": 0

}

## Update

URL :

- POST http://localhost:90/api/teams/

Example :

{

"name": "Team A",

"skills" : "javascript,java,html,nodejs",

"members" : "member1,member2,member3,member4"

}

## Delete

URL :

- DELETE http://localhost:90/api/teams/

Example :

{

"id": "5bfe26b1b509011b709f6501" // This is automatically assigned when the project is created

}

# Bids

Any team member can Bid but only one bid is accepted per Team. Teams should coordinate to make their selection and then any team member can send the vote.

## List

URL :

- GET http://localhost:90/api/bids/

Example Response :

[

{

"\_id": "5bff51fdbcec4035144372be",

"projectID": "5bf395c91e53214ea0b80615",

"teamID": "5bfe2be542c48127fc5d5e6f",

"\_\_v": 0

}

]

## Get Bid by ID

URL :

- GET http://localhost:90/api/bids/{id}

Example Response :

{

"\_id": "5bff51fdbcec4035144372be",

"projectID": "5bf395c91e53214ea0b80615",

"teamID": "5bfe2be542c48127fc5d5e6f",

"\_\_v": 0

"order" : 1

}

## Create

URL :

- PUT http://localhost:90/api/bids

Example Request :

{

"projectID": "5bf395c91e53214ea0b80615",

"teamID" : "5bfe2be542c48127fc5d5e6f",

"order" : 1

}

## Update

URL :

- POST http://localhost:90/api/bids

Example Request :

{

"projectID": "5bf395c91e53214ea0b80615",

"teamID" : "5bfe2be542c48127fc5d5e6f"

"order" : 1

}

## Delete

URL :

- DELETE http://localhost:90/api/bids

Example Request :

{

"id": "5bfe26b1b509011b709f6501"

}

# Features

There are a set of features for every project.

## List

URL :

- GET http://localhost:90/api/features/

Example Response :

[

{

"\_id": "5bff73b20abd600c943b2f2e",

"projectID": "5bf3959b1e53214ea0b80613",

"description": "Users should be able to add users",

"title": "Add user",

"status": "open",

"\_\_v": 0

},

{

"\_id": "5bff74320abd600c943b2f34",

"projectID": "5bf3959b1e53214ea0b80613",

"description": "Users should be able to add users",

"title": "Add Accounts",

"status": "open",

"\_\_v": 0

},

{

"\_id": "5bff74440abd600c943b2f35",

"projectID": "5bff6dbb3342ce29601cc093",

"description": "Users should be able to add users",

"title": "Add Accounts",

"status": "open",

"\_\_v": 0

},

{

"\_id": "5bff750500f606340486aedf",

"projectID": "5bff6dbb3342ce29601cc093",

"description": "Users should be able to add users",

"title": "Hello World",

"status": "open",

"\_\_v": 0

}

]

## Get Feature by ID

URL :

- GET http://localhost:90/api/features/{id}

Example Response :

{

"\_id": "5bff73b20abd600c943b2f2e",

"projectID": "5bf3959b1e53214ea0b80613",

"description": "Users should be able to add users",

"title": "Add user",

"status": "open",

"\_\_v": 0

}

## Get Features by Project ID

List all the features for a specific project.

URL :

- GET http://localhost:90/api/features/project/{id}

Example Response :

{

"\_id": "5bff73b20abd600c943b2f2e",

"projectID": "5bf3959b1e53214ea0b80613",

"description": "Users should be able to add users",

"title": "Add user",

"status": "open",

"\_\_v": 0

}

## Create

URL :

- PUT http://localhost:90/api/features

Example Request :

{

"projectID": "5bff6dbb3342ce29601cc093",

"description" : "Users should be able to add users",

"title" : "Hello World",

"status" : "open"

}

## Update

URL :

- POST http://localhost:90/api/features

Example Request :

{

"projectID": "5bff6dbb3342ce29601cc093",

"description" : "Users should be able to add users",

"title" : "Hello World",

"status" : "closed",

"id" : "5bff73c90abd600c943b2f30"

}

## Delete

URL :

- DELETE http://localhost:90/api/features

Example Request :

{

"id": "5bfe26b1b509011b709f6501"

}