

# AFFORDABLE

## Technical Documentation

Andrew Hamiel, Nick Miller, John Spurney, Lexus Uhlir  
CSE 5911 / Summer 2019 / Julia Armstrong

### Table of Contents

Table of Contents

Features Added

- MFA

- User verification/Recovery

- Account Management

- User/Administrative Management

Tools Used

Architectural Diagram

Code Organization

- Component Information:

Future Features & Some Carry-Over Bugs

- Application Center > Apply

- Application Automation

- Registration

- Forgot Username or Password

- Design

# Features Added

## MFA

- I. Created Two-Factor table in Database
- II. QR Codes generated in application
- I. Allow users to register device for Multi-Factor Authentication via Google Authenticator
- I. Allow users to register multiple devices
- I. Allow users to remove a Multi-Factor device from an account
- I. Text field on Settings page that lists Multi-Factor devices

## User Verification/Recovery

- I. Users can register in two ways
  - a. Full registration “Apply Now”
  - b. Basic information only “Browse Dashboard”
- II. Existing bug with changing password on user recovery fixed
- III. Allow users to create custom recovery questions
- IV. Password protected accounts with recovery options
  - a. If MFA is enabled, then it is required for password recovery
  - b. If MFA is not enabled, a verification email is sent to a user
- V. Add unverified emails to table, allow users to verify them from Settings page

## Account Management

- I. Users allowed to have multiple emails on file
  - a. Ability to delete primary email disabled
  - b. Ability to delete secondary emails
- II. Allow users to change recovery questions
- III. Allow users to change their password
- IV. Email sent to user when password is
  - a. Changed
  - b. Forgotten and being recovered
- V. Email sent to user when an email is
  - a. Added
  - b. Modified
  - c. Deleted
  - d. Changed to primary
- VI. Emails redesigned to be more aesthetically appealing to users
- VII. Ability to deactivate account
- VIII. Automatic logout after 15 minutes

## User/Administrative Management System

- I. All pages except for user registration/login blocked unless user successfully logs in
- II. Leveraged API that stores IP addresses
  - a. Can also determine State, City, etc.

- III. Recent profile modifications listed on settings page
  - a. Login, password change, etc.
- IV. Username listed in navbar to improve personalization
- V. Profile page contains enhanced information on emails
  - a. Primary indicator
  - b. All emails on file
  - c. Verified/unverified status
- VI. Read receipt pixel database changes made
- VII. Bootstrap SweetAlert used for alerts to improve UI aesthetics

## Testing

- I. Test script to create test data
  - II. UI Testing - Selenium
    - A. Register account → browse dashboard and apply now
    - B. Login → success and failure
    - C. Email
      - 1. Add email
      - 2. Change primary email
      - 3. Delete email
    - D. Change security questions
    - E. Change password
  - III. Unit Testing - Mocha
    - A. Email
      - 1. Add email
      - 2. Change primary email
      - 3. Delete email
      - 4. Proper email validation
    - B. Password
      - 1. Change password
- Tests can be run inside of the server directory by using the command 'npm test'

# Tools Used

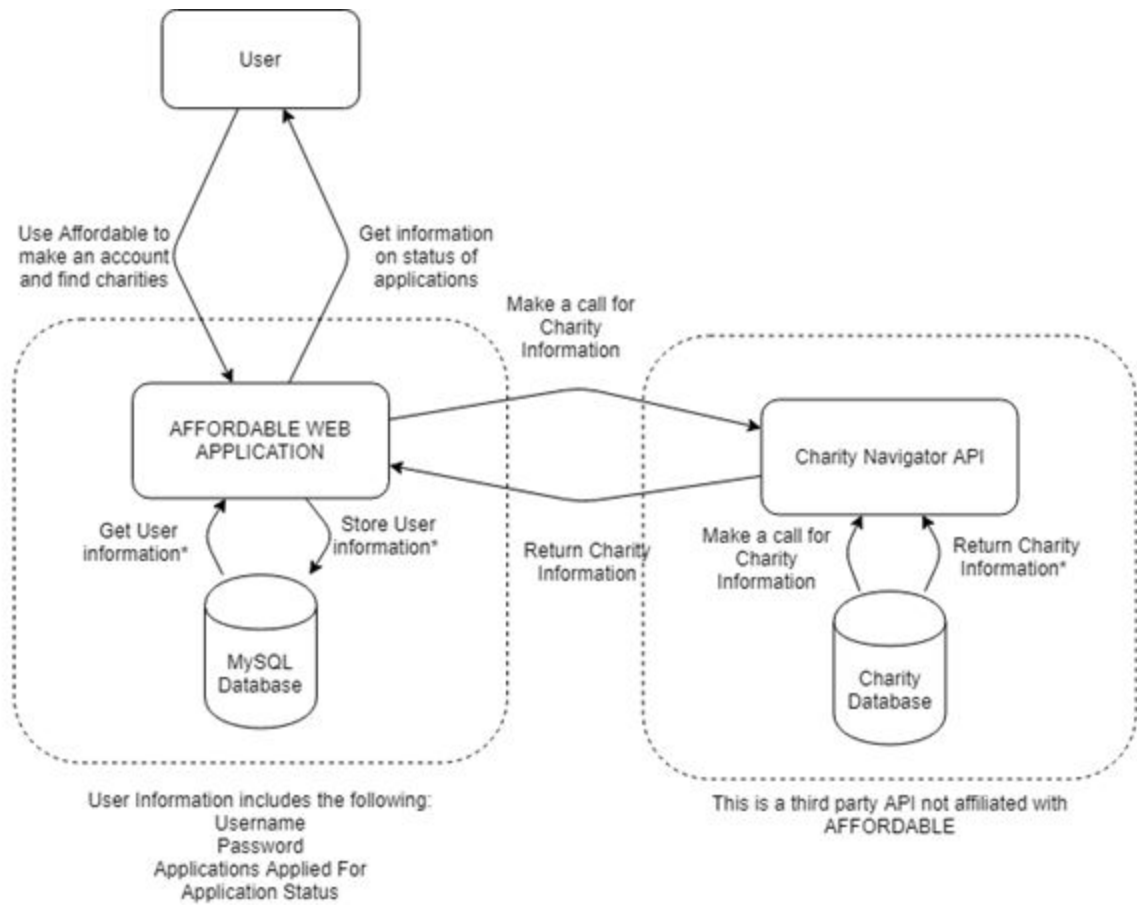
- JavaScript
  - Node JS
  - React
  - Express JS
- Charity Navigator API
- CSS/HTML
- MySQL
- Selenium
- Mocha

## **To download any required libraries:**

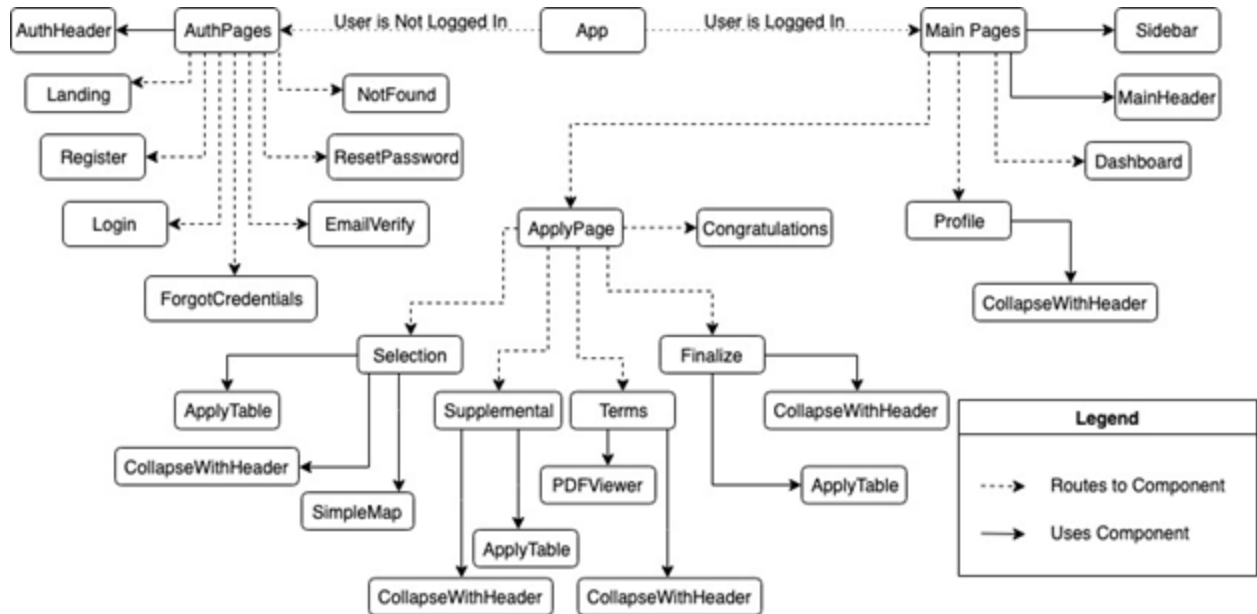
1. Navigate to affordable folder in terminal/console window
1. Run

```
> npm i
```

# Architectural Diagram



# Code Organization



### Component Information:

- App
  - Directs User to AuthPages if Not Logged In
  - Directs User to MainPages if Logged In
- AuthPages
  - Renders the AuthHeader on all pages
  - Renders component depending on current route
- AuthHeader
  - Header for all AuthPages
- Landing
  - Landing Page of the Website
  - User can choose to Login or Register
- Register
  - Page where user can register an account
- Login
  - Page where user can login
- ForgotCredentials
  - Page where user can retrieve forgotten credentials
- EmailVerify
  - Page shown when user verifies email
- ResetPassword
  - Page that allows user to change their password

- NotFound
  - Page that is shown when route a user has entered does not exist
- MainPages
  - Renders the Sidebar and MainHeader on all pages
  - Renders component depending on current route
- Sidebar
  - Navigation bar for all MainPages
- MainHeader
  - Header for all MainPages
- ApplyPage
  - Container for Selection, Supplemental, Terms, Finalize, and Congratulations Pages
  - Renders component depending on route
- Selection
  - Page where users can search for charities and find programs they would like to apply to
- Supplemental
  - Page where users can upload required documentation for a program and answer any supplemental questions
- Terms
  - Page where users agree to the Terms and Conditions of Affordable and any program they are applying to
- Finalize
  - Page where users submit their applications
- Congratulations
  - Page that pops up when users successfully submit their application
- ApplyTable
  - Reusable Table Component for Apply Pages
- CollapseWithHeader
  - Reusable Component whose contents can be hidden from view by clicking the header of this component
- Simple Map
  - Component that uses Google API to show location of chosen charity in Selection Page
- PDFViewer
  - Reusable Component that shows a PDF
- Profile
  - User information that will be used for application of programs
- Dashboard
  - Application status manager
    - Page break
- Settings
  - Add email

- Change email
- Delete non-primary email
- Change primary email
- Change password
- Add multi-factor device
- Remove multi-factor device
- Change security questions

## Future Features & Some Carry-Over Bugs

### Application Center > Apply

- I. Selection Page
  - A. Registered programs functionality
  - A. Application list functionality
- I. Supplemental Page
  - A. Required documentation functionality
  - A. Supplemental questions functionality
- I. Terms Page
  - A. Agree/disagree terms and conditions validation
  - A. Organization terms and conditions functionality
- I. Finalize Page
  - A. Submit application functionality

### Registration

- I. The way the password is stored in the database needs to be revisited. Past team believes that it is not being properly encrypted

### Design

- I. Read receipt pixel needs to be implemented on emails sent to users
- I. Login successful etc. pop-ups could be cleaner and are sometimes unnecessary