

INFSCI 2560 WEB TECHNOLOGIES & STANDARDS

Final Project: Admission Portal

15th April 2018

Team Members:

Prashant Wankhede (PRW19@pitt.edu)

Lei Bao (LEB129@pitt.edu)

Bowen Xu (BOX3@pitt.edu)

Author: Prashant Wankhede (prw19), Lei Bao (leb129), Bowen Xu (box3)

Date: 15th April 2018

Admission Portal is a 'MEAN STACK' application

#To run you need

1. MongoDB

2. NodeJS

Then,

On Windows:

1. run mongod from mongo install directory and keep it in background

2. cd to this project directory here in cmd prompt and run,

node app.js

3. Open Google Chrome to access front-end at

<http://localhost:3000/>

4. Create an account either by registering directly or through social websites. All

features of the websites are functional. Delete node_modules directory in case of conflict and run again

Steps should be similar on MacOSX or Linux based environment

#Technology and Process Description:

- \$ Admission Portal uses live textual data present on <https://core.letsenrol.com/api/partner-colleges-universities>. This link can be changed to fetch any other country's data by transforming respective variables such as exam type, thus making it a portable application. The aim is to make a website that takes dynamic updates presented via a server as mentioned in final project description of this course
- \$ Login and register function on Admission Portal uses passport API's that take respective 'secret' keys from Google, Facebook, and Twitter. The cookie calculator is simple jwt function that uses secret key 'anamika' to generate cookie value expiring in 24 hours. Note - Today's websites such as Microsoft employs a much better implementation of cookies/unique keys to deal with 100 Million requests each day and the same functionality can be extended to Admission Portal once the user base reaches that goal
- \$ The college data can be filtered at a real-time based on the variables on left of collage page extracted from the data received from the server. The data loading integrates Angular loading bar at the top as the get function fetches data from the server into local MongoDB
- \$ The collage comparator is able to compare up to 3 colleges simultaneously. Comparing depends upon the available data variables fetched by the server. If there are no courses listed on the respective college drop-down menu, the compared values do not populate but the same information can be eventually gathered by clicking more college information button from the collage page
- \$ The animation presented on homepage uses simple JavaScript that takes classes such as fade in left, fade in right or zoom in to show a bunch of information to the user but at the same time presenting website's motive and how it can help to prospective students in a brief display
- \$ The register page and contact page use Nodemailer module of NodeJS which in turn uses Sendgrid API with username 'prashaw' (credentials owned by Prashant) to send emails from the local. The same functionality can be extended into various other functions such as subscribe to a newsletter or refer a friend or export history to emails or send/receive application reports to universities to pretty much anything a standard website does to control email-based requests

#Future work:

To make the application more scalable, database technology could be changed to DynamoDB which takes care of millions of requests. Additionally, Node provides the ability to create multiple v8 instances of node using node's child process (`childProcess.fork()`) each requiring 10mb memory as per docs) on the fly, thus not affecting the main process running the server. So, offloading a background job that requires huge server load becomes a child's play and we can easily kill them as and when needed. Nevertheless, Admission Portal is a ready to deploy node-based project on any web server with minimal effort or customization

=====