Project Report

SDDA – 27 March 2024

By: Quinlan Caiger

(design, development process, challenges faced, and solutions implemented)

Web Application Design

Web pages/components

There are 8 total web pages/components (represented by unique HTML templates/files) each associated with a unique route. They all consist of the same header and footer (as of right now they don’t share it with the “base.html” file but hopefully they will).

Navigation

Each of the HTML templates are rendered via an associated route. The links for each of these pages are stored in the navigation bar to allow the user to move between pages. The application also automatically moves between pages upon certain user interaction for example, after signing up a user will be transported to the movies page.

Development Process

The first obstacle that had to be overcome was determining a name for the company that the car insurance application was going to be developed for. A trustworthy yet interesting name was desired, without copying an established company’s name. After much procrastination and frustration, the name was taken from a personal favourite brand of cologne “Cool Water”.

The names of web pages/components were then determined. These pages were inspired by existing insurance applications and determined with the project scope in mind. The project then began development.

Steps

* Created a new local project folder
* Created a local environment (check README.md for details)
* Initialised a new GitHub repository (check README.md for details)
* Initialised a new Flask application (check README.md for details)
* Created blank HTML files for each web page/component
* Set up routes for each of them and tested them by rendering the associated HTML template in the browser
* Created “base.html” to share the header and footer amongst all pages at a later stage
* Started developing each of the “smaller/simpler” web pages/components moving to larger/more complex ones
* Created some “dummy” local data to be displayed
  + User data
    - {id, name, email, pic, policy-id}
  + Employee data
    - {id, name, job-title, pic, desc}
  + Policy data
    - {id, name, price, poster, desc}
* For pages/components that require local dummy data template syntax/moustaches “{{ }}” were used to access this data and display it
* In addition, where a list of data (usually dictionaries) had to be displayed, “JINGA” template syntax was used for improving the developer’s experience and improving readability and decreasing code size/repetition
* CRUD operations were then created for the user and policy data
  + These were then tested using Postman
* CSS was then added for each of the developed pages to improve styling/presentation
* After reaching a point where the “about”, “policies”, and “policy” pages had been created, the “help” page was developed taking inspiration from the [“contact us” page on the MiWay insurance web application](https://www.miway.co.za/contact-miway)
* The footer was then created with help from this [internet resource](https://www.w3schools.com/howto/howto_css_social_media_buttons.asp)
* After Monday’s workday the CSS of a few pages was changed and then the profile, login and signup pages were developed

Challenges faced and solutions implemented

The challenge of “starting”/having some basic plan for the web design and development process was solved through drawing inspiration from existing web applications and upon the knowledge we have acquired throughout the course.

The challenge of ensuring code implemented worked was solved through testing (using Postman and hot reloading in the browser) and thorough debugging based on the examples we had implemented during the course. In addition, successfully passing data from routes to the HTML templates was achieved through looking at previous work examples and debugging/trial and error.