Coursework 2 Report
Online Shift Tracker

Igors Ahmetovs
40125689@napier.ac.uk
Edinburgh Napier University - Advanced Web Technologies (SET09103)

Abstract

This report describes the "Online Shift Tracker" web application that was produced as a coursework deliverable for Advanced Web Technologies module. The report summarises the main functionality of the application and gives justification to the design rationale behind it. It also looks at potential improvements that could be made in the future.

Keywords – Python, Flask, Web, App, Jinja, Shifts, Salary

1 Introduction

Overview The produced application has been created to aid potential users in tracking the time they spend at work and estimate their monthly earnings. Because of the nature of shift work, especially when a regular shift pattern is not present, it might be difficult to have full awareness of ones income, as this is something that might greatly vary on a monthly basis. A solution to this would be to keep track of all the shifts and store the information in an accessible space.

The Shift Tracker is an online application that supports multiple users. After logging in, a person has access to their calendar, populated with the shift they have previously entered into the app. By initially specifying the hourly wage to the program, it will calculate the expected earnings for the month, calculate the total number of shift and hours. This will Provide a user with data that might aid them in the process of ensuring their pay was accurate.

Days of the month that have a shift scheduled/worked are clearly identifiable on the calendar and display the hours of the shift for better clarity. Upon clicking on a particular day, the user will have the opportunity to either add a new shift for that given day, or, if the day already has a shift scheduled - amend it or completely delete it.

The data within the web application is persistent as it is supported by a database running on the server, thus saving every interaction of the user.

2 Design

The initial stage of the development process was the careful planning and design of the website architecture. It was highly important to develop a solid structure that would efficiently underline the functionality of the Shift Tracker.

The users interaction with the web application is through a collection of URL's that are carefully routed in the code. Based on the passed URL variables, a single defined page could serve multiple purposes.

The "/login" page provides the facility to login into the website or create a new account in the database. The "/" or the route page is then used to present the current month to the user if no variables have been passed. By specifying a year and a month as part of the URL, the month representation will be of that month rather than the current month's one (e.g. /2017/11 will print the visual representation for November 2017). This result can also be achieved in a more simple way - there are "arrow" hyperlinks provided for the users to browse through the months.

After proceeding to the next page by either clicking on the specific day of the month or passing an additional ¡day¿ variable into the URL, the user will see a different page, printing the details of the shift on that particular day. The form fields will be pre-populated if a shift record exists for the day, if not, they will be left empty. Apart from the shift details, a "Delete" button will also be present if the day has a shift scheduled.

The data persistence is achieved by deploying an SQLite database on the server. The users actions (that are executed by form submissions) are then fed into the database to take the appropriate actions, such as retrieve or update data.

3 Enhancements

The user's experience with the Online Shift Tracker application could be improved by enhancing the visual aspect of the website and the user's interaction with it. The web app could benefit from a Single Page Application approach - this would mean that all the data would be loaded into a browser, and the web app can be dynamically populated and navigated without any page refreshes. This would achieve a similar experience to Google Calendar, which imitates the behaviour of a native app, rather than a collection of individual pages in a browser.

By utilising CSS frameworks, such as Bootstrap, a bit more, the front-end could become more visually appealing.

By taking on the points above, and also presenting the user with instant, responsive feedback (e.g. by using JavaScript), the web application would provide a much more dynamic and interactive experience.

In terms of functionality, it would be a big improvement if the Shift Tracker could also account for overtime shifts that are paid at a different rate. Paid holiday support would also be a nice addition.

4 Critical Evaluation

The general website structure is well designed - the URL are logical and can be easily recognised as dates - thus adding more transparency to the user and another way to interact with the web app's logic.

The duplication of data is minimised by using the functionality offered by Flask.

5 Personal Evaluation