Web Application Development Coursework 2 Report

By Robert Helps

https://rhelps25.pythonanywhere.com

Why this website

When I was deciding what kind of website to develop, I had one goal in mind. A website that could be used by a wide range of people which is why I chose this idea. Everyone likes to listen to music: all ages, genders and nationalities. I then decided to create a website where a user could add and view their records to a database, which I believe I developed successfully.

Web forms

One use of Flask forms in my application is the User form. The use of this form is to get user data from the user into the User database. In this form we collect the users Username and Password, ensuring both have data inputted. This form is used when logging in as well as creating a new user. Perhaps in the future I could use two different forms, so that I could collect different types of user data when logging in and registering the user.

Another form we use is when adding a record to the database. In this form we record the Artists name, genre as well as the records name. These pieces of information are recorded into two separate databases which are connected by an association table. The use of this method is used due to the many to many relation between the Artist and Album.

Database

Another key feature of my web application is the use of a database to store information through sqlalchemy. The use of a database is used to store user data from the forms implemented. The User database stores the user data (username and password) as long with a unique id which is used as the primary key of the table. We also verify if the user exists in the database in the login and register route as we do not allow repeat usernames in the database.

Another two more database tables are used in my model. One of these databases (Record) stores the record and other key information such as Artist and release date. This table also contains a unique id for each record which we will use as a primary key of the table. Moreover, we have created a many to many relationship between the Record database and the user database. However, in flask to implements this type of relationship we must use an association table to link the Record and User tables.

In order to use this many to many relationship, the use of the association table is used which holds the unique id of an object in the Record and User table in the form of foreign keys. In order to access this table and therefore form a relationship between the databases, I have created a relationship in Record which tells the user by the ‘Owner’ and how this owner owns a ‘record’.

Authentication and sessions

With every website, authenticating users is vital to ensuring security and preventing exploitation of the web application. In my web application, I use flask-login to login and logout a user. Some pages are blocked off to the user unless they are logged in. I chose to implement this feature as it does not make sense to log out if you are not logged in. A user not logged in can also not view records as they are seen to have no records unless logged into their account.

In my application, the use of session is implemented once. When a user logs into my website, their unique id is saved into the session with key “id”. This is so that the id of the user is known in other views. We use the id from the session when adding a record so that the association table saves it to that user as well as when a user views their records. When a user logs out, the id key is deleted as we will no longer need it in the current session.

Styling

When styling my web application there was one goal in mid, to have a professional looking website. I first decided to style my website using external CSS in the static folder which is accessed in the base html template.

When styling my forms, I decided to have a light blue background. This decision was made so that the user is initially drawn to the form where data is submitted but is not too out of place compared to the whole page. Furthermore, I have also decided to have a box around each submission box, this is to make the form look neat. However, in a future iteration, I would like to test to see how the submission boxes would look if it stretched at 100% width.

Another key use of styling in my application is the styling of flashed messages. A flash message is thrown by the view to a template when a key action is performed. I decided when styling these messages to show them to the user in a box similar to what an alert would be. I decided to do this in order to grab the user’s attention and tell them what the web application is doing.

Unit testing

When designing my web application, I decided to use a test-driven approach through the use of unit testing. There were a few key unit tests that I decided to implement as I felt they were most vital for a successful deployment of the application.

The first set of unit tests I preformed was to test database queries and how data is stored in the database. The use of database testing was important to ensure user data is stored correctly and securely. Furthermore, queries into the database were tested to ensure the correct data is fetched when displaying through the views.

Another set of unit tests used was ensuring the forms worked as intended. I decided that these sets of tests would test the boundary cases and performed the correct validation of the submissions. This was done in order to collect the correct data from the user and to store suitable data in the database.

One final set of tests used was ensuring that pages loaded and the correct html code was returned. When html receives a request, it returns a specific code to tell the developer what happened. I decided to use these codes to ensure that my application did what it was designed to do. In addition, the tests ensured that the user would not have an experience they would not expect.

Logging

Logging was another feature that I decided to integrate into my web application. Logging is used in order to track what is happening in the framework. To implement logging, I decided to setup a layout for what is logged into the file record.log. The layout consisted of recording: the time of the log, level

of the log, name of the log, the thread name and the message displayed by the log. I feel this format is best for tracking what is logged and how it could be tracked efficiently

A few key parts of the application are logged. One of the types of logs is when something is inserted into a database. When inserting into the database, we issue an info log and the message includes what is inserted and any other information that may be needed. Another type of log used is when an error occurs when submitting data. When data that can be inserted into the database is received, an appropriate log is created in order to record the error.

Deployment

To deploy my web application, I chose to do it through pythonanywhere, this was because I was recommended due to its ease of deployment. Moreover, throughout the development of my application, I chose to use version control through git. With the continuous use of git, I was able to rollback and add features easily. When I finally decided to deploy me website, I used the services of git clone to quickly add all my source code to pythonanywhere. With all my code set up in a virtual environment I deployed my web application to the URL linked on the first page.

Other features

In the html templates of my web application other features such as Jquery, Javascript and HTML geolocation have been used. The Use of Javascript has been used to show and hide media in the homepage. This has been chosen as not all users will want to see the media and just care about the use of the webpage.

I have also decided to implement html geolocation of the website. The use of geolocation has been used to show the user their coordinates, in a later implementation of my website this could be used to show local, independent record stores as well as concerts in the area.

3 tier architecture

It is common for many web applications to now implement a 3 tier architecture. These tiers consist of a presentational, Business logic and Data access. The presentational tier in my application is used through the forms collecting data as well as the html templates used which extends from the base page. Furthermore, in my application the use of a Business logic tier is implemented through the use of routes. This can be shown in views.py where each route renders a templated for a html page as well as all the logic that needs to be displayed on that page. Finally, a Data access tier is implemented through the use of my database model as shown in models.py. My database model can be accessed by the tier above in order to query data and display it in the presentational tier.

Accessibility

In my web application, I designed it from the ground up to be accessible to as many users as possible. One way in which I have done this is by choosing a colour scheme that people with colour blindness would not find hard to use. The use of this colour scheme will allow the website to be used by many types of users. In addition, I have made the website with a simple and readable layout. This would allow users with technical difficulties the ability to use my website, without too much difficulty.

Security

One of the most common types of security vulnerabilities in web development is the improper use of authentication and session management. In my application, I use the flask-login library in order to properly authenticate a user when they log in. In my application, a user must be logged in to use certain features such as viewing records, this could be seen as protecting user data from potential hijackers. Furthermore, when a user logs out, their session id is deleted so a potential hijacker cannot get this information. Another key protection of user data that I could implement in the future is hashing a user's password. This will further secure a user's most precious piece of data with us.

Another type of security concern addressed is Insecure Direct object reference. This vulnerability happens when an attacker obtains a reference to an internal object. To address this, in my application, every form had data validation checks to prevent this kind of attack.

References

Record shop, shown on the homepage: [Ellie Hodgson; 2022]

Bunch of records, shown on the homepage [Square Deal Recordings & Supplies; 2022]