Qixuan Wu

Render: https://flaskproject3.onrender.com

1. Introduction

The purpose of this report is to outline the design and development process of a web application that displays video game sales data. The application includes two pages: a Game list page that lists games based on their sales data, and a Game sale page that displays detailed sales data for a selected game.

2. Design

The overall design of the application is simple and clean, with a focus on usability and ease of navigation.

2.1 Game list page

The Game list page displays a table of the ranked by their sales data. Each row of the table contains information about a single game, including the game's rank, name, platform, year of release, genre, and publisher.

2.2Game sale Page

The Game sale page displays detailed sales data for a selected game. The page includes a table that lists the game's sales data for various regions. The game's title is displayed at the top of the page, with a hyperlink to return to the Game list page.

3. Development

The application was developed using the Flask web framework, with HTML and CSS for the front-end design. The data for the application was sourced from a CSV file containing sales data for video games, which was processed using Python to create the necessary data structures.

3.1 Game list Page

The Game list page was designed using HTML and CSS, with the Flask framework used to dynamically populate the table with data from the CSV file. The table is generated using a for loop that iterates over the data, creating a new row for each game. The hyperlink for each game's title is created using Flask's url_for function, which generates a URL based on the name of the view function that handles the request.

3.2Game sale Page

The Game sale page was also designed using HTML and CSS, with the Flask framework used to dynamically populate the table with data from the CSV file. The data for the selected game is passed to the view function as a parameter, which is then used to populate the table. The hyperlink to return to the Game list page is created using the same url_for function as in the Game list page.

4. Render Deployment

The application was deployed using Render. The deployment process involved creating a new Render service, connecting it to a GitHub repository containing the application code, and configuring the necessary settings for the Flask framework and dependencies. The final deployed version of the application can be accessed at [insert deployment URL].

5. Conclusion

Overall, the design and development of this video game sales application provide easy access to detailed sales data for thousands of video games. The use of Flask, HTML and CSS allowed for the dynamic generation of the pages, enabling easy updates and modifications to the data and design. The application was successfully deployed using Render, providing a reliable and scalable platform for users to access the application.