CP 1895 Project – Flask Web Application ( 20 % final mark)

The project is to create a web app to display a video game collection. (You may substitute video games for something which you are more interested in… for example Hockey Cards, Art Work, Action Figures, movies…etc)

The project must use the following technologies:

Python Flask server to generate the html pages for the project.

Jinja2 templates to re-use html

Javascript for form validation

CSS for html styles

SQL Lite database to store information about the collection

Git and Github for version control

Optional Technologies (for a higher mark, please see rubric)

Deploy to Amazon AWS ( you will need to have a working login system to manage access before deploying)

Bootstrap to manage styles

Research and add 1 additional technology to your web app

The Project must have the following functionality:

1. Homepage introducing the collection.
2. Page where collection is displayed. Each collectible item displayed must include at least one image and at least 3 text elements. For example, in my video game collection, one game would include the following:

* Image for Game
* Console
* Year released
* Description of game

1. Form to upload new collection items to the server. Need to have both javascript validation for form and server side validation of data. Need to handle security related to upload of image file to only allow image format and to prevent malicious files.
2. Page to remove collection items.

Optional functionality (for a higher mark, please see rubric)

Add a login system. A user will be restricted from adding or removing items if they are not logged in. ( you need this feature if you plan to deploy to AWS)

Project Deliverables:

The project will be split up into 2 deliverables. The Project Plan will be worth 3% of your final mark and the final project submission 17%.

Project Plan (3 % of final mark) (Due Friday Nov 5th)

For the project plan, please submit the following:

1. Word document describing your web app. Please include the following.

* Listing and description of each separate page on your web app.
* Description of which part of each page will be part of a template and which part will be generated dynamically
* Explain how navigation will work on your web app (which links will be available on each page)
* Which level on the rubric you are aiming for

1. Word document describing technical requirements of your application. Please include the following:

* Listing of all database tables. For each table, include the SQL Lite command used to create the table.
* Listing of any optional technologies you plan to include
* Do you plan to deploy to AWS?
* Listing of all static files (html, css, images) you plan to use with your web app
* Additional Resources. Go and find 3 additional resources and list them here. These can be tutorials, or webpages which you will use to learn more about creating web apps with flask.

Final Project Submission (17% of final mark) (Due TBA, early December)

For the final submission, you will submit your github url and a document explaining the steps to setup and run your web app. If you have deployed to amazon aws, you will also submit the live url for your web app.

Please see the following Rubric for how the final submission will be evaluated:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Visual Presentation | Git/Github | Functionality | Extra Features | Flask Setup | Bugs |
| 90 – 100% | Simple presentation, using CSS to control layout and style | There is a consistent record of commits to the github account on a weekly basis, since the start of the project | All basic functionality is present to allow the collection to be viewed, new items to be added and items to be removed | All extra features are implemented (login system, aws, bootstrap, and one additional feature of your choice) | Proper use of flask features (templates, jinja2, routes) | A few minor bugs |
| 80-90% | Simple presentation, using CSS to control layout and style | There is a consistent record of commits to the github account on a weekly basis, since the start of the project | All basic functionality is present to allow the collection to be viewed, new items to be added and items to be removed | Login feature added | Proper use of flask features (templates, jinja2, routes) | A few minor bugs |
| 70-80% | Simple presentation, using CSS to control layout and style. A few issues present with layout | There is a consistent record of commits to the github account. A few gaps present where student was not committing work | All basic functionality is present to allow the collection to be viewed, new items to be added and items to be removed | No extra features | Proper use of flask features (templates, jinja2, routes) | Several small to medium bugs exist |
| 60-70% | Simple presentation, using CSS to control layout and style. A few issues present with layout | Several large gaps in commit history. | All basic functionality was attempted, however there are parts of the data management (add new items, remove items which do not function correctly) | No extra features | Proper use of flask features (templates, jinja2, routes) | Several medium severity bugs exist in addition to a major bug with data management |