

COMPSCI 235 S2 2023

Assignment Phase 3

15% of your grade

Due date: 11:59 pm Thursday 21st September

Total – 35 marks

Objective

The purpose of Assignment phase 3 is to add new features to the Flask web app you built in Assignment phase 2. The focus will be on adding features that improve the user experience, such as user authentication, search, and pagination. You will also improve the app's quality by adding more tests. The goal is to make the app more interactive and functional.

Each team member is required to demonstrate their individual contribution as a lack of participation will result in a 0 mark.

Any GitHub repository to which you link will be interpreted as your own work.

IMPORTANT: For your assignment to be marked, it must adhere to the following requirements:

It must be functional and meet the requirements.

Non-Functional Requirements

1. Project Structure and Requirements [1 mark]
 - Conformance to the project structure and requirements.
 - Ensure vital documentation, like `requirements.txt` and `README.md`, is present.
 - The application should run by typing `flask run`, from within the virtual environment in a terminal window.
 - The tests should run by typing `python -m pytest -v tests`, from within the virtual environment in a terminal window.
 - If the application doesn't run properly, the remaining steps will NOT be marked.
2. Use of Blueprints [1 mark]
 - Register blueprints to segregate the application's components. For example: home, games, genres, search etc.
 - Each blueprint should encapsulate a distinct set of functionalities adhering to the principle of Single Responsibility.

3. User Interface & Design [1 mark]

- Implement CSS for a visually captivating application.
- Utilize Jinja templates to ensure a consistent appearance across pages while minimizing repetitive code.

4. Flask Routes and HTTP Protocol [2 marks]

- Clearly define routes/URLs for different functionalities (e.g., /games, /genres, or /search etc.)
- Ensure that the application uses the appropriate HTTP methods, query parameters, response codes etc.

5. Abstract Repository Interface [2 marks]

- This interface should define the necessary methods for fetching, storing, and manipulating data about games.
- For this assignment, you'll be using a memory repository as the data storage mechanism.

Functional Requirements (NEW!!!)

Task 1: User Authentication

- Implement a registration form with a password validator
- Provide clear feedback for users who are logged out.
- Ensure the use of decorators for route protection.
- Users once logged in, should be able to access routes. You'll implement the `@login_required` decorator. The `@login_required` decorator should be implemented using `login_required()` from `flask_login`.

Task 2: User Reviews – Overall

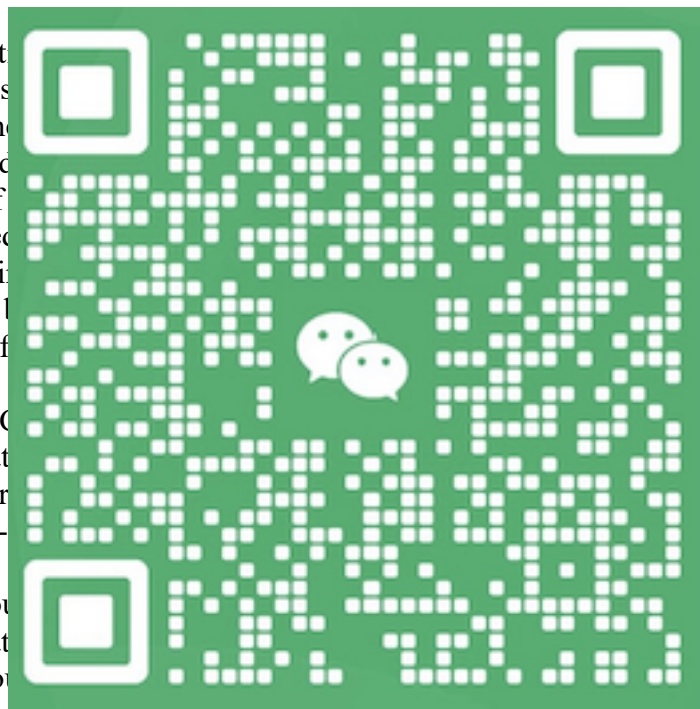
- Allow authenticated users to leave reviews.
- Reviews and average ratings should be displayed on the game page. Display these reviews and ratings on the game page using relevant criteria.

Task 3: Wishlist or Favourites

- Allow authenticated users to add games to their wishlist.
- The wishlist/favourites should be displayed on the user's profile page or a separate dedicated section within the web application.
- Allow users to remove games from their Wishlist or Favourites list.

Task 4: User Profile Page [5 marks]

- Design and implement a user profile page that is accessible only to authenticated users. This profile page should display an organized history of the user's activities, such as games they've rated, reviewed, or added to their wishlist.
- This task mostly involves creating a new view that collates and displays existing data relevant to each user.
- Implement CSS styling to make the user profile page visually appealing. The design should be consistent with the rest of the application but can have unique elements that make the profile page special.



Task 5: Unit Tests for New Features [3 marks]

- Extend the unit tests to cover all new functionalities introduced.
- It is important that the tests are written well and provide good coverage. These should be readable with meaningful names, and include comments detailing the test cases.

Task 6: End-to-End Testing [3 marks]

- Implement end-to-end tests to simulate complete user requests, such as registering, logging in, browsing games, adding games to the wishlist, logging out etc.
- It is important that the tests are written well and provide good coverage. These should be readable with meaningful names, and include comments detailing the test cases.

Submission:

- For marking, we will use the latest commit that you have pushed to the GitHub repository before the submission deadline. Please make sure your submission is pushed to the remote GitHub repository, and not just committed to your local version of the repository!
- Each webapp should display the name of the developers (the team members).
- When assessing your work, markers will expect to:
 - Find a README file in the repository that explains how to set up a virtual environment, how to run the application, and how to run the tests.
 - Run the application in a virtual environment in a terminal window.
 - Run all tests in a virtual environment.
 - **Failure to follow these instructions will result in a failure to mark the assignment.**

⚠ Sharing assignment solutions is against the academic integrity policy and is not allowed. It is not acceptable to share solutions or code in your assignment submission. Therefore, please do not share your assignment solutions, or code, or assignments or solutions with anyone. If you do, your solution. There are also consequences for what is legitimate and what is not.

frequently, our leading to solutions, ion of this will result disciplinary action. ce code leading to st not publish meone copies your ave any doubt over

You can refer to online tutorials, generative-AI tools (such as ChatGPT) and online resources. However, please learn from them and implement the solutions yourself based on what you learnt from these sources. You must not copy from these sources.

Don't leave your computers, devices, and belongings unattended — you must secure these at all times to prevent anyone having access to your assignments or solutions. If others are found to have used your solution, you will also face disciplinary action.