Course Name: Computing with IT Management

Year/Group: 4th Year ITM

Semester: 7

Module Title: 4th Year ITM Project

Lecturer Name: Enda Lee

Steam Dashboard Overview and Project Feature Prototypes

Student Name: Ricardo Danganan Jnr

Student Email: x00191395@mytudublin

Submission Date:

Due Date:

Table of Contents

Steam Dashboard Overview

This project will create a comprehensive gaming dashboard for Steam users, combining features from game optimization, game management, community interaction, and analytics. It will serve as a single platform for users to manage their entire gaming experience—ranging from game recommendations and performance monitoring to price tracking and community engagement.

Key Features

To bring together all the elements of the previous ideas, you can break the project down into the following main features:

Game Library Management and Recommendations:

- Game Library Overview: Display all games in the user's Steam library, organized by tags, genres, and playtime.
- Recommendations System: Analyse user preferences to recommend games based on playtime, genres, or friends' favourites.
- Wishlist Price Monitoring: Track price changes of games in the user's Wishlist and provide historical price data.

Game Performance Monitoring and Optimization:

- System Performance Monitor: Track real-time CPU, GPU, and RAM usage while playing games.
- FPS Tracking and Optimization Suggestions: Monitor frames per second (FPS) for games and provide optimization tips based on the user's system specifications.
- Optimizing System for Installed Games: Use data from game requirements to suggest optimal settings for each game

Achievements Tracking and Analytics:

- Achievement Overview: Display all unlocked and locked achievements for games in the library.
- Goal Setting and Progress Analyzer: Allow users to set goals for unlocking achievements and track progress towards those goals.
- Leaderboard Integration: Compare achievements with friends.

Community Engagement and Friends Interaction:

- Friend Activity Insights: Track the games friends are playing and their status.
- Session Planner: Allow users to schedule co-op gaming sessions based on friends' availability and shared games.
- Trending Games & News Aggregator: Display trending games, developer news, and gaming events relevant to the user's library.

Game Library Analytics:

- Playtime Insights and Analysis: Provide visual insights into which games the user plays most frequently, their favourite genres, and the time of day when they play.
- Backlog Analyzer: Identify underplayed or never-played games in the library and make suggestions on what to play next.

Technologies to Use

1. Frontend: HTML, CSS, JavaScript, with frameworks like React.js for building a user-friendly interface.
2. Backend: Node.js or Flask for handling Steam API calls, data processing, and storing user preferences.

3. Database: Supa-base or another SQL database for storing user information, game analytics, Wishlist, etc.
4. APIs:
    - Steam Web API for gathering game data, user achievements, and friends' activities.
    - Price Tracking APIs to monitor game prices.
    - Authentication: Steam OpenID for secure user login.

Project Phases

Phase 1: User Authentication & Game Library Integration

- Set up Steam OpenID authentication.
- Pull user game data and display the game library.
- Start building the basic UI of the dashboard.

Phase 2: Performance Monitoring and Optimization

- Integrate Open Hardware Monitor or custom code for gathering system specifications.
- Build modules to collect CPU, GPU, and RAM data and match with each game's requirements.
- Provide optimization recommendations.

Phase 3: Achievements and Community Engagement

- Display achievements from Steam API and allow users to set and track progress.
- Show friend activities and add features like scheduling game sessions

Phase 4: Wishlist Price Monitoring

- Track prices for games in the user's Wishlist using price APIs.
- Provide notifications or suggestions when prices drop.

Phase 5: Game Library Analytics and Final Integration

- Build visual analytics for playtime, achievements, and backlog analysis.
- Integrate all the features into a cohesive dashboard experience.
- Enhance user experience by adding a navbar, visual elements, and intuitive navigation.

Additional Ideas for Improvement

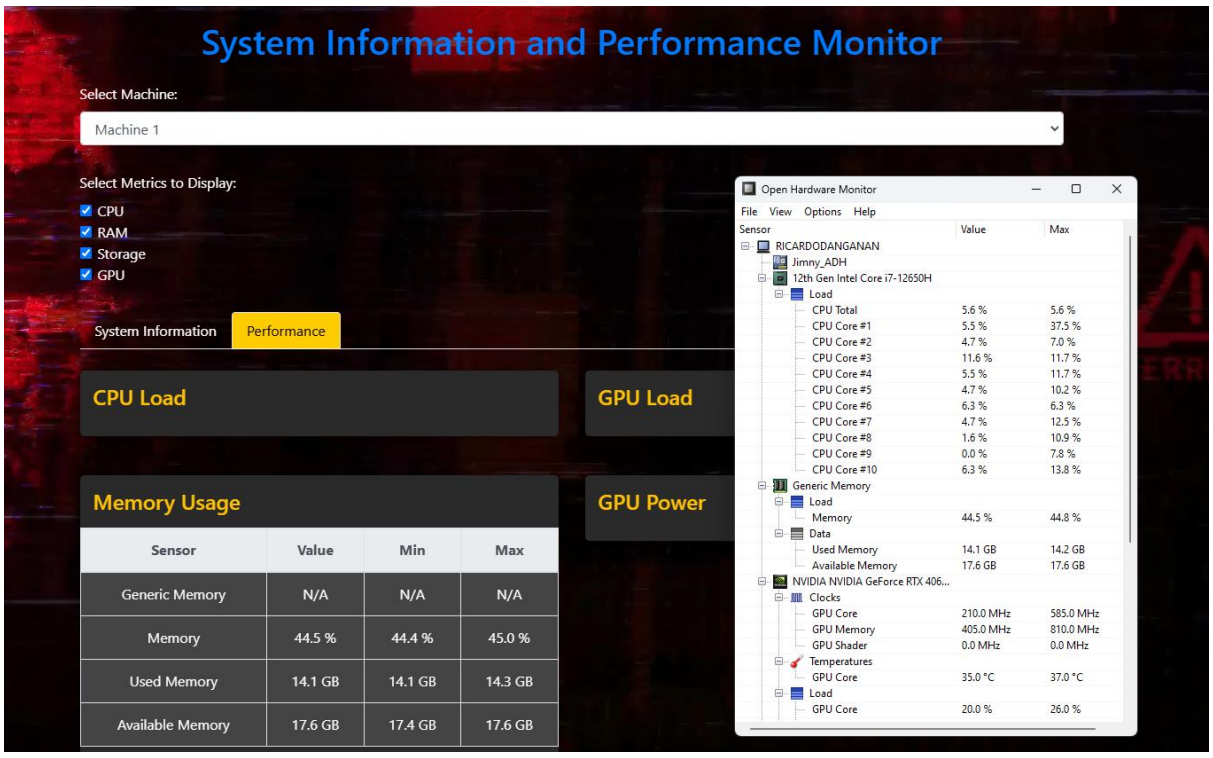
- Cross-Platform Compatibility: Ensure that the dashboard works on both Windows and Linux systems.
- Customization Options: Allow users to customize their dashboard theme, colours, and layout.
- Notifications: Integrate notification features for price drops, friend activities, or achievement goals.
- Cloud Backup: Store user preferences and progress in the cloud, allowing for access across multiple devices.

Project Challenges & Considerations

- API Limitations: Some Steam APIs might have rate limits; you may need caching to avoid being blocked.
- Real-time Data Handling: Collecting real-time system data can be challenging; optimization and performance testing will be needed.
- Data Privacy: Handle user data securely, especially when tracking personal preferences, playtimes, and hardware data.

Prototype 1: Performance Monitoring Feature (using 3rd party tool)

The first prototype of the Ultimate Steam Gaming Dashboard focuses on **real-time performance monitoring** using **Open Hardware Monitor**. It provides insights into system metrics like CPU, GPU, RAM, and temperature, enhancing the gaming experience by identifying potential system bottlenecks.



Implementation:

Tools:

- **Frontend:** HTML, CSS, JavaScript.
- **Backend:** Node.js.
- **Third-Party Tool:** Open Hardware Monitor API.

Features:

- Fetches real-time metrics (CPU/GPU usage, RAM, temperatures).
- Displays data using simple graphs and tables.
- Updates every 5 seconds for near real-time insights.

Challenges:

- **API Limitations:** Some hardware metrics depend on system compatibility.
- **CORS Issues:** Resolved with browser extension during local testing.
- **System Overhead:** Minor impact on performance, especially for low-end PCs.

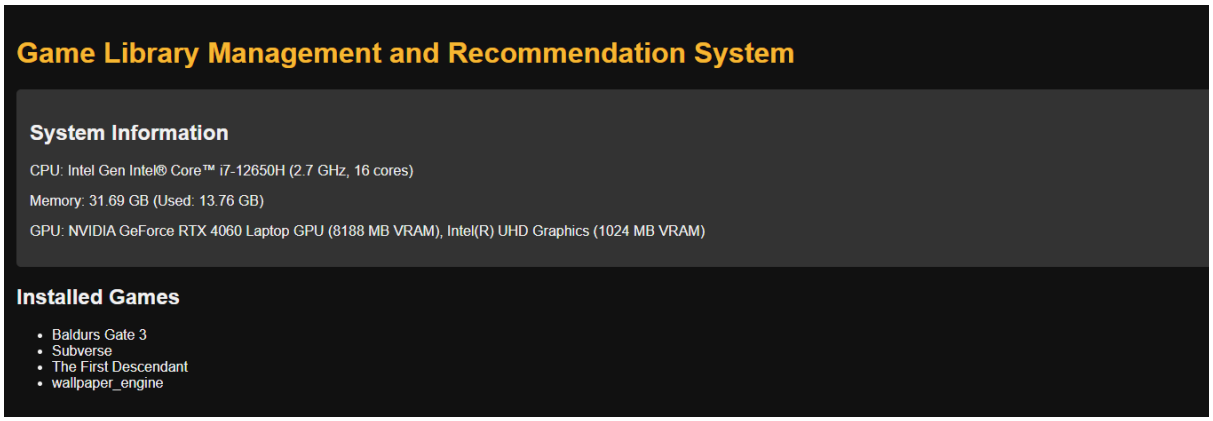
Achievements:

- Successfully retrieved and displayed live performance data.

- Created basic visualizations for CPU, GPU, and RAM usage.
- Demonstrated integration feasibility with the dashboard.

Prototype 2: Built-in Hardware Detection for Game Optimizer Feature of my Project

For the second prototype of the Game Optimizer, I successfully implemented a self-contained hardware detection system using a Node.js backend. This eliminates the need for external tools like Open Hardware Monitor and provides users with real-time system specifications directly on the dashboard. The current functionality supports displaying essential hardware information such as CPU, GPU, and memory details.



Key Features:

Hardware Detection:

- Fetches CPU, GPU, and memory details via Node.js and displays them on the dashboard.

Game Detection:

- Detects installed games locally without Steam API integration.

Frontend Display:

- Displays system specs and game list in a clean, user-friendly interface.

Challenges:

- Simulating Open Hardware Monitor was overly complex, so I simplified the approach using the system information library.
- Steam API integration was avoided due to private API restrictions and security concerns.
- Considering PowerShell scripts for future networked system monitoring.

Achievements:

- Created a fully self-contained solution for hardware detection.
- Simplified the system by avoiding database and Steam API dependencies.
- Enabled dynamic game and hardware data retrieval for end-users.

Future Plan:

- Expand metrics to include disk usage and network activity.
- Integrate PowerShell scripts for network monitoring via SSH.

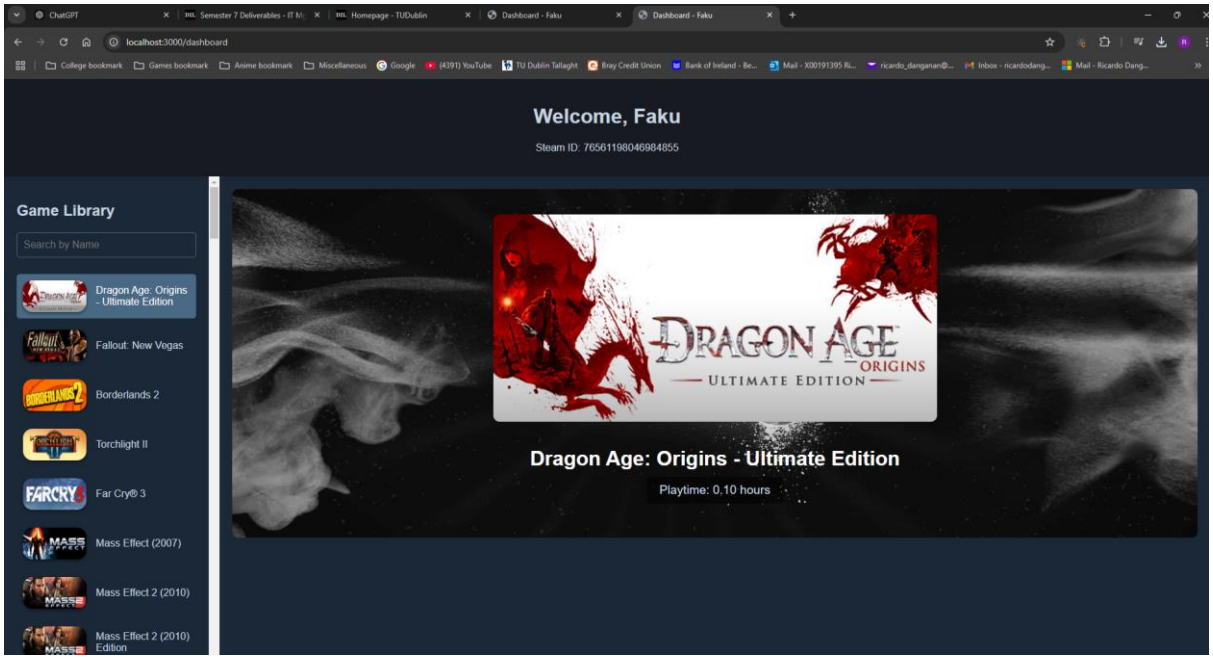
- Provide game optimization recommendations based on hardware specs.
- Enhance the dashboard's UI for better usability.

Conclusion:

This prototype successfully delivers built-in hardware detection and game listing features, laying the foundation for future expansions like network integration and optimization recommendations.

Prototype 3 - Authentication & Game Library Fetch

The third prototype introduces user authentication and Steam integration, allowing users to log in with their Steam accounts and access their game library in the dashboard. This prototype enhances personalization by using Passport.js for Steam authentication and Express.js for backend development.



Features:

- Integrated Steam OpenID authentication via Passport.js.
- Utilized Axios to fetch game library details from the Steam API.
- Seamlessly redirected users from the login page to the game library view (/dashboard) after successful authentication.

Impact on User Experience:

This new feature adds a personalized experience for users, directly integrating their existing Steam library into the platform, making it truly a gamer-centric dashboard.

Future-Plans:

- Adding community engagement features, such as friend recommendations.
- Enhancing game analytics to offer users suggestions for improving performance.
- Implementing further optimization features for installed games.

Screencast URL:

https://tudublin-my.sharepoint.com/:v:/r/personal/x00191395_mytudublin_ie/Documents/IT%20Management%202024-2025%20Files/Project%20Files/Project%20Proposal%203/X00191395-Project-Screencast.mp4?csf=1&web=1&nav=eyJyZWZlcnJhbEluZm8iOnsicmVmZXJyYWxBcHAiOiJPbmVEcml2ZUZvckJ1c2luZXNzIiwicmVmZXJyYWxBcHBQbGF0Zm9ybSI6IldlYiIsInJlZmVycmFsVW9kZSI6InZpZXciLCJyZWZlcnJhbFZpZXciOiJNeUZpbGVzTGlua0NvcHkifX0&e=0qcMzi