A Project Report On

Game Infomation System

Submitted in partial fulfillment of the requirement for the award of the degree

Bachelor of Computer Application
BCA

Academic Year 2025 – 26

Krish Kachhadiya
92300527035

Akshit Baleliya
92300527101

Jay Kakadiya
92300527099

Internal Guide
Prof. Kinjal Raval

Rajkot-Morbi Road, At & PO : Gauridad, Rajkot 360 003. Gujarat. India.

Marwadi University
Marwadi Chandarana Group

NAAC A+

Faculty of Computer Applications (FCA)

Certificate

This is to certify that the project work entitled

Game Information System

submitted in partial fulfillment of the requirement for the award of the degree of

Bachelor of Computer Application

BCA

of the

Marwadi University

is a result of the bonafide work carried out by

Krish Kachhadiya (92300527035)

Akshit Baleliya (92300527101)

Jay Kakadiya (92300527099)

during the academic year 2025-26

Prof. Kinjal Raval	*Dr. Sunil Bajeja*	*Prof. (Dr) Sridaran Rajagopa*
Faculty Guide	**HOD**	**Dean**

DECLARATION

I/We hereby declare that this project work entitled **Game Information System** is a record done by me.

I also declare that the matter embodied in this project is genuine work done by me and has not been submitted whether to this University or to any other University / Institute for the fulfillment of the requirement of any course of study.

Place : Rajkot

Date : 03/09/2025

Krish Kachhadiya (92300527035) Signature : KRISH

Akshit Baleliya (92300527101) Signature : AKSHIT

Jay Kakadiya (92300527099) Signature : JAY

CONTENTS

SYNOPSIS

Project Title: Game Information System – A CSV-Based Game Information Using Python and Tkinter.

This project presents GUI application developed in Python to manage and organize information related to video games. Designed with simplicity and efficiency in mind, the system allows users to perform key operations such as adding, updating, searching, deleting, and clearing game records. Data is stored using the lightweight CSV format, eliminating the need for complex database management.

PREAMBLE

General Introduction:
The video game industry is growing fast, and many people enjoy keeping track of the games they own or play. However, using notebooks or spreadsheets to store this information can be slow, messy, and hard to manage.

The **Game Management System** is a simple yet efficient GUI application developed in Python to manage a collection of video games. It utilizes CSV (Comma-Separated Values) files for data storage instead of a traditional database, making it lightweight and portable.

Module Description:
The system is organized into the following functional modules:

1. **Add Game:**
   Allows the user to enter details such as game name, type, platform, release date, developer, total downloads, version, rating, minimum age requirement. Each game is saved in the CSV file.

2. **Display Game:**
   Displays all stored Game in a tabular format by reading from the CSV file.

3. **Update Game:**
   Enables the user to change the details of an existing game entry. Users can skip fields using the Enter key

4. **Delete Game:**
   Enables the user to remove a particular game from the CSV file based on the game names.

5. **Search Game:**
   Lets the user search for specific games by game name.

6. **Sort By Rating:**
   Displays all games sorted by their rating from highest to lowest.

7. **Filter Games By Platform:**
   Allow users to filter and display games based on specific platforms like PC, Mobile, etc.

8. **Get Highest Rated Game:**
   Automatically highlight and display the game with the best rating

Each module is designed to be user-friendly, efficient, and focused on the needs of gamers and collectors who want a reliable and quick way to manage their game library without using complex data or online platforms.

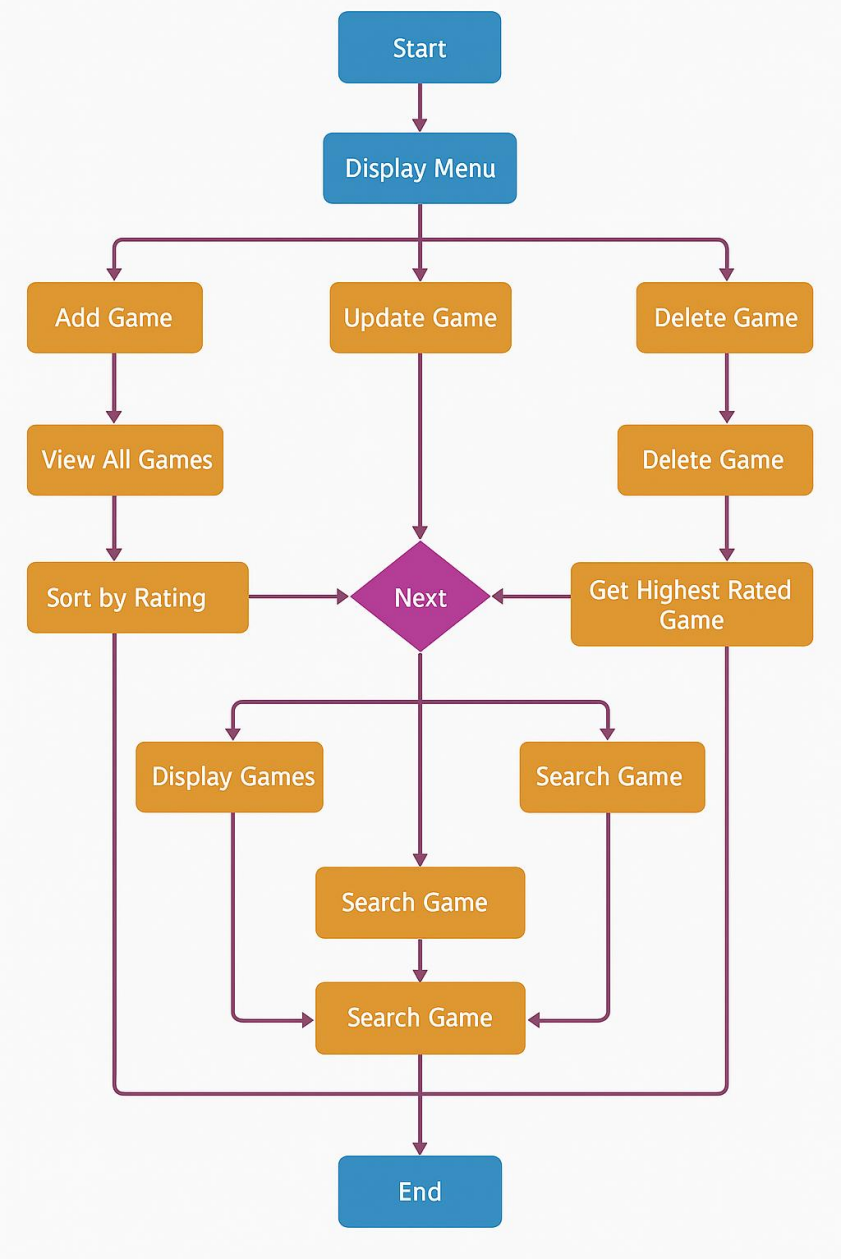
TECHNICAL DESCRIPTION

- **Hardware Requirement:**
  - Any system capable of running Python
  - At least 512MB RAM (minimal requirement)
  - Minimal disk storage for CSV files

- **Software Requirement:**
  - Python 3.x (recommended)
  - Any text editor or IDE for code modifications
  - Graphical desktop environment to run the Tkinter GUI application
  - No command-line interaction is required, but Python should be executed in a shell or terminal to launch the app

SYSTEM DESIGN AND DEVELOPMENT

1. Flowchart:



[Flow chart of Game Information System]

2. Use Case Diagram:

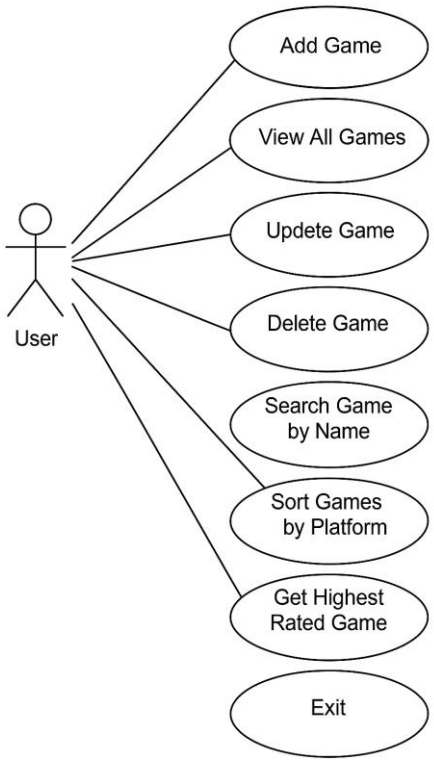


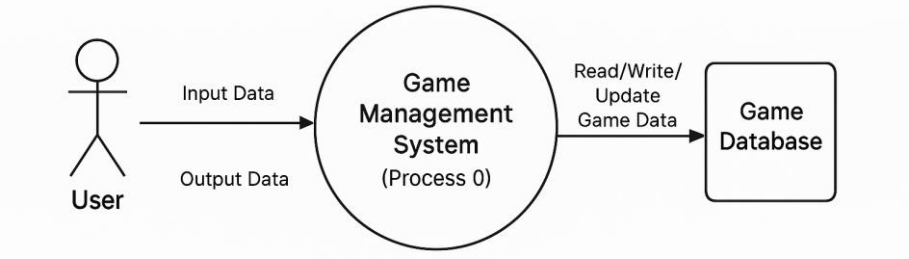
Figure: Use Case Diagram of Game Information System

The use case diagram illustrates the interaction between the user and the core functionalities of the Game Management System. A single actor, User, interacts with the system through nine key use cases:

- **Add Game:** Allows the user to input a new game, including details such as name, platform, and rating**.**
- **View All Games:** Displays a list of all games currently stored in the system.
- **Update Game:** Enables the user to modify the details of an existing game.
- **Delete Game:** Removes a specific game from the system using its identifier.
- **Search Game by Name:** Searches for a particular game based on its name.
- **Sort Games by Platform**: Filters the game list to show only games from a selected platform.

This diagram helps visualize the user-driven functionalities in the system.

3. Data Flow Diagram – Level 0:



The Level 0 Data Flow Diagram provides a high-level overview of the Game Management System. It illustrates the interaction between the **User** and the **central system module**, where user requests such as adding, viewing, updating, deleting, searching, sorting, and filtering games are processed.

4. Data Flow Diagram – Level 1:

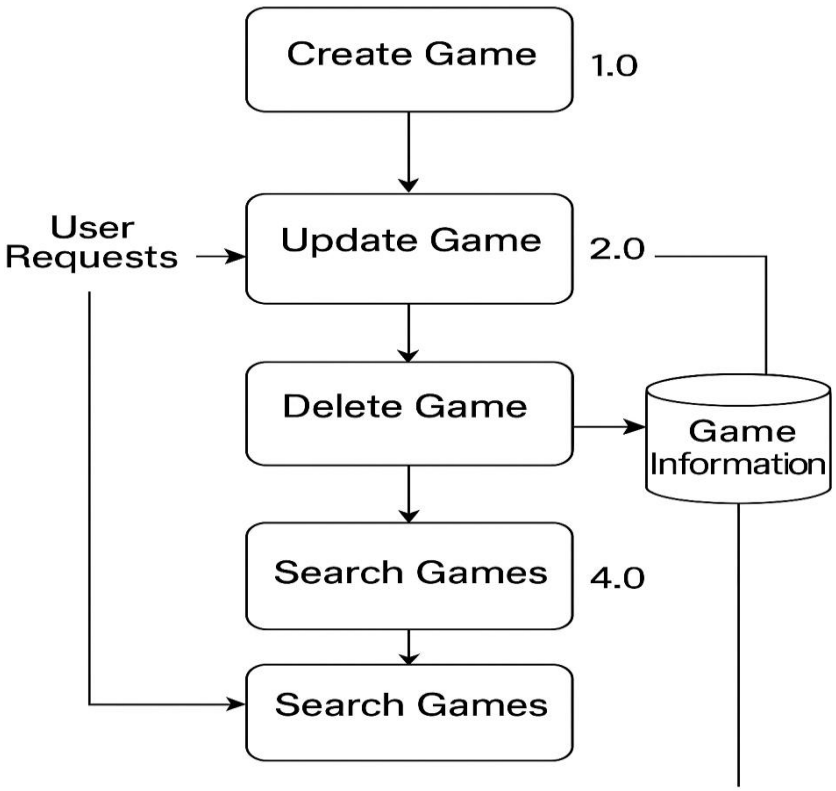


Figure: Level 1 Data Flow Diagram of Game Management System

The Level 1 DFD breaks down the main system into four specific processes:

- **1.0 Add Game** – Accepts new Game entries and stores them.
- **2.0 Search Game** – search game by their names.
- **3.0 Update Game** – Modifies specific fields of a selected Games.
- **4.0 Delete Game** – Deletes a Games record from storage.

All modules interact with the **Games** data store (CSV file). The **User** acts as the external entity initiating all processes. This diagram helps in understanding the internal modules and their data flow more precisely.

5. Class Diagram

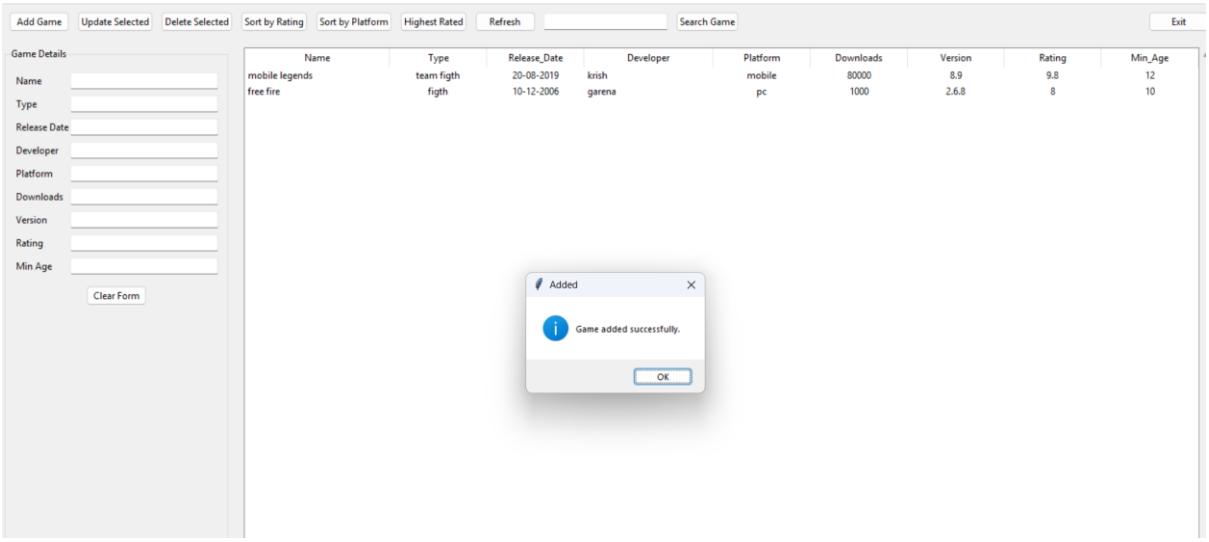
	A	B	C	D	E	F	G	H	I	J
1	name	type	release_da	developer	platform	downloads	version	rating	min_age	
2										
3										
4										

6. Screen Design & Coding

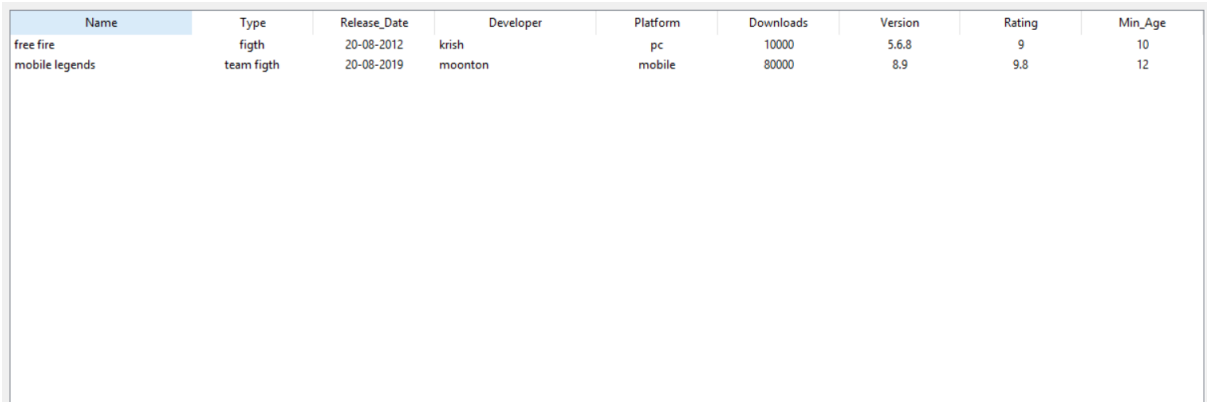
Screen Design (CLI)

The project uses a text-based interface displayed on the command line. Upon running the program, the user is presented with a numbered menu containing all major functionalities:

Adding a Game

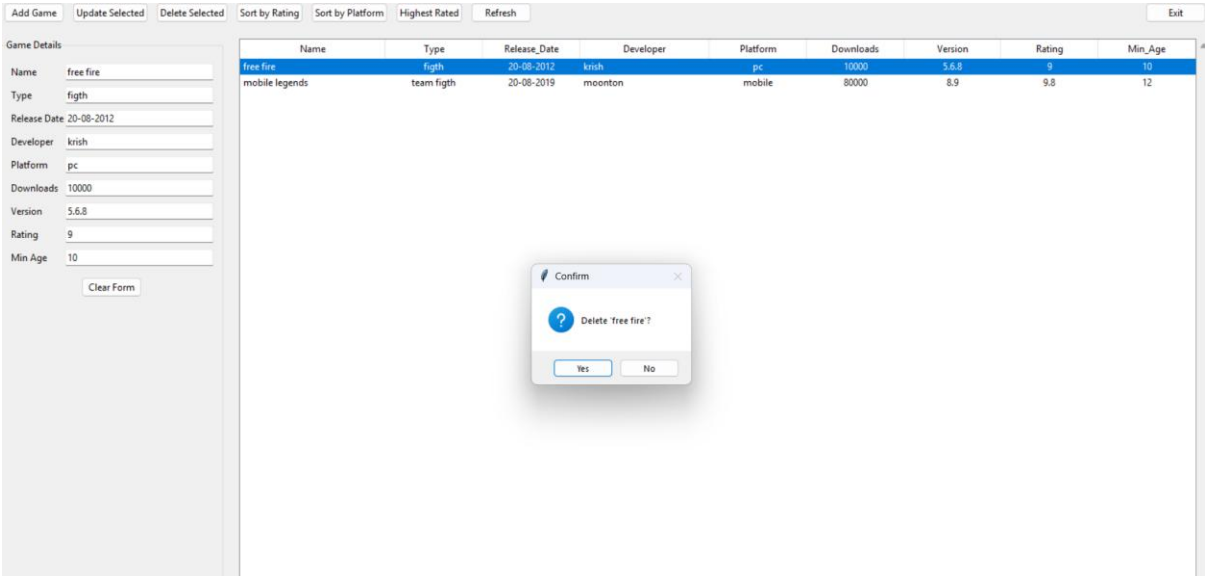


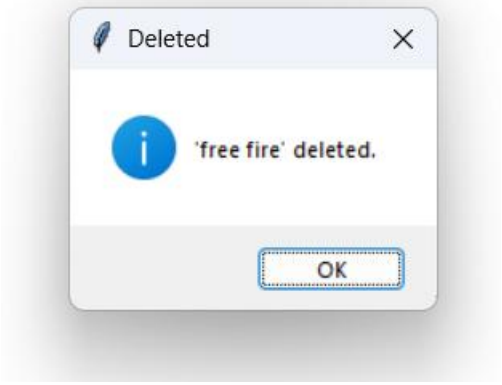
Display Game



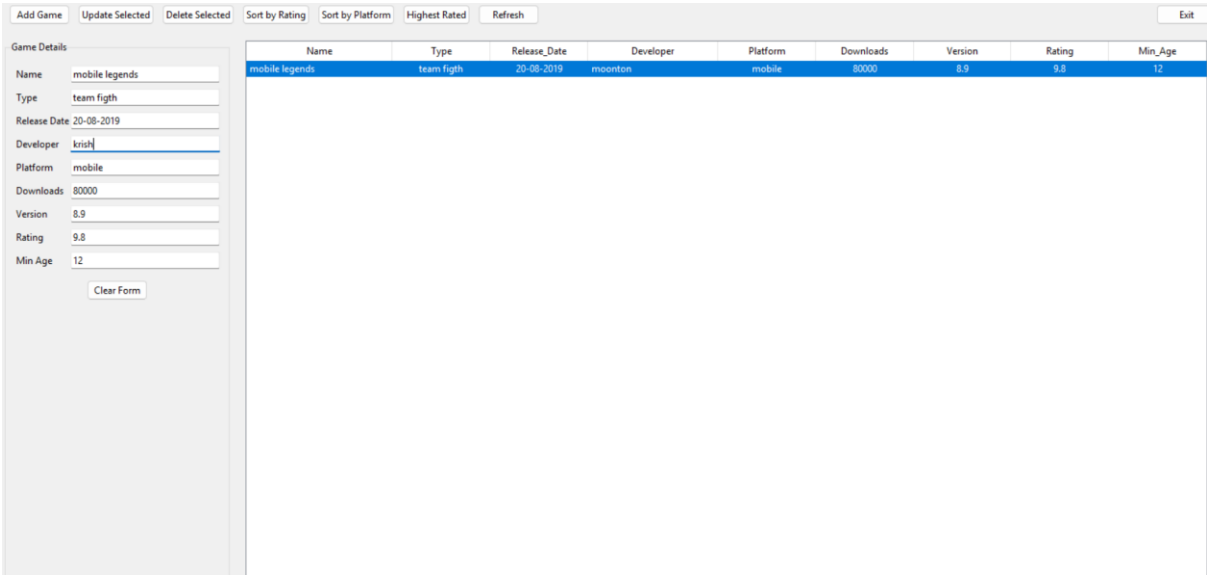
Name	Type	Release_Date	Developer	Platform	Downloads	Version	Rating	Min_Age
free fire	figth	20-08-2012	krish	pc	10000	5.6.8	9	10
mobile legends	team figth	20-08-2019	moonton	mobile	80000	8.9	9.8	12

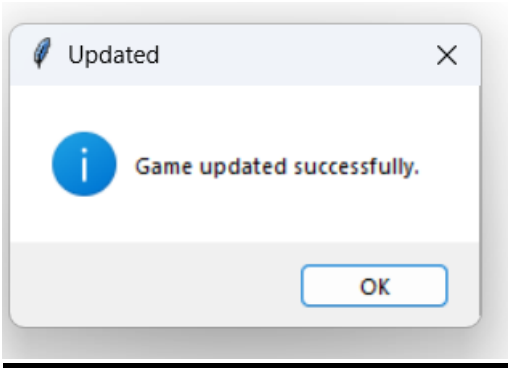
Delete Game


Add Game
Update Selected
Delete Selected
Sort by Rating
Sort by Platform
Highest Rated
Refresh
Exit
Game Details
Name free fire
Type figth
Release Date 20-08-2012
Developer krish
Platform pc
Downloads 10000
Version 5.6.8
Rating 9
Min Age 10
Clear Form
Name Type Release_Date Developer Platform Downloads Version Rating Min_Age
free fire figth 20-08-2012 krish pc 10000 5.6.8 9 10
mobile legends team figth 20-08-2019 moonton mobile 80000 8.9 9.8 12
Confirm
Delete 'free fire'?
Yes
No

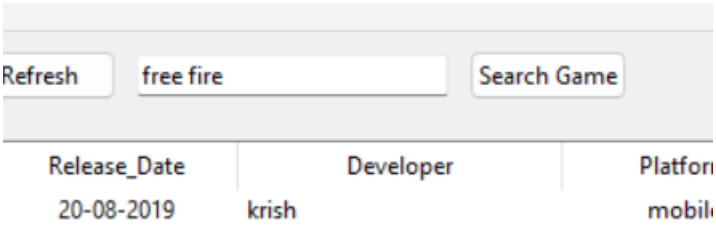

Deleted
'free fire' deleted.
OK

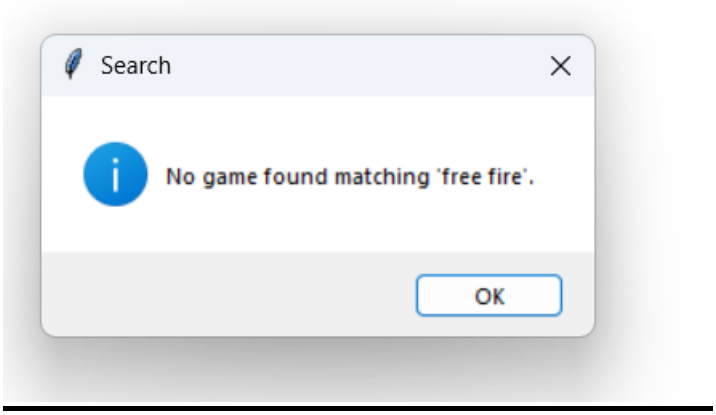
Update Game


Add Game
Update Selected
Delete Selected
Sort by Rating
Sort by Platform
Highest Rated
Refresh
Exit
Game Details
Name mobile legends
Type team figth
Release Date 20-08-2019
Developer krish
Platform mobile
Downloads 80000
Version 8.9
Rating 9.8
Min Age 12
Clear Form
Name Type Release_Date Developer Platform Downloads Version Rating Min_Age
mobile legends team figth 20-08-2019 moonton mobile 80000 8.9 9.8 12

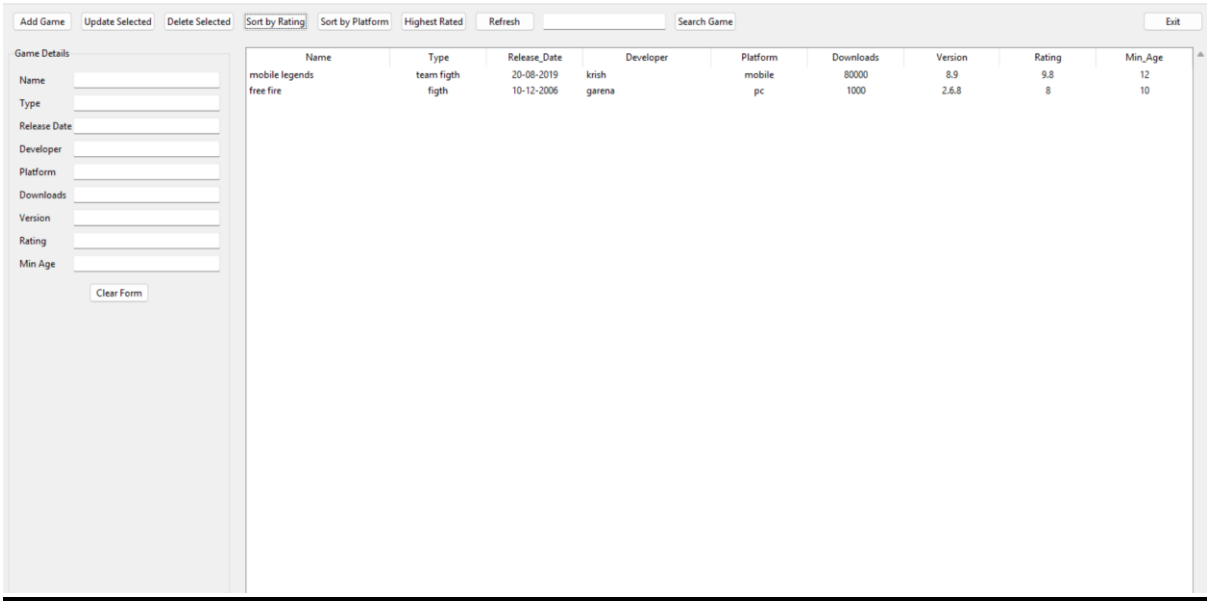

Updated
Game updated successfully.
OK

Search Game

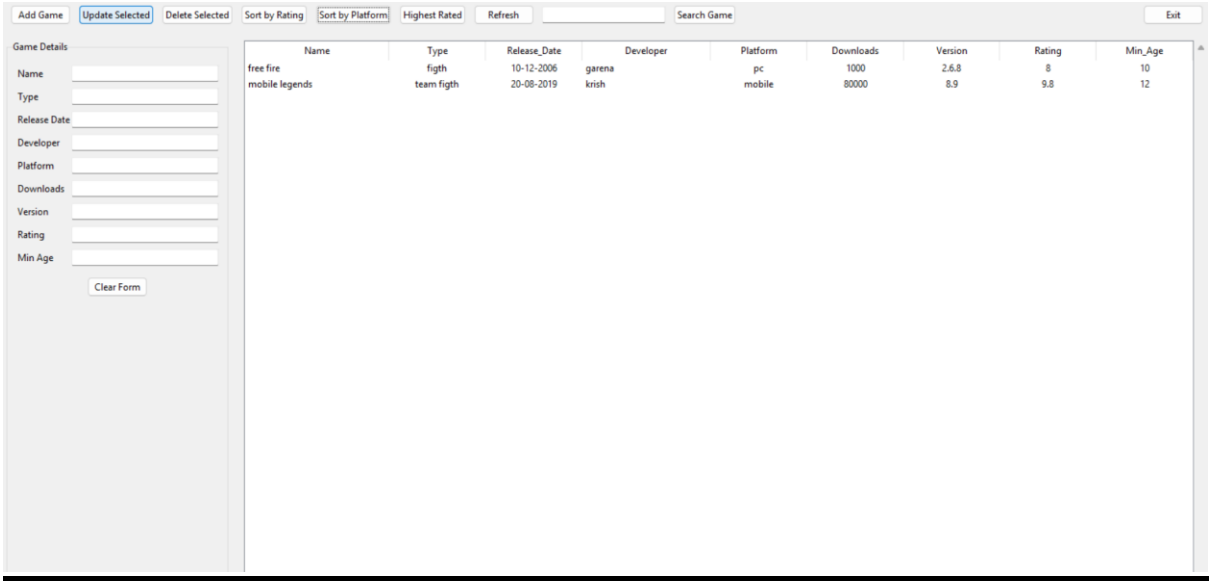

Refresh
free fire
Search Game
Release_Date
Developer
Platfor
20-08-2019
krish
mobil


Search
No game found matching 'free fire'.
OK

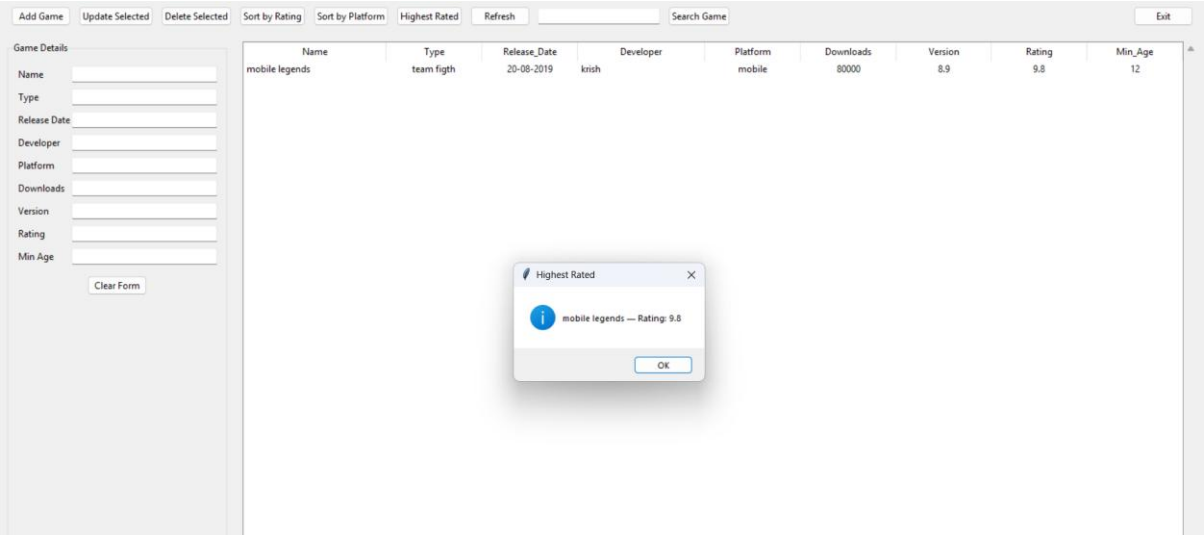
Sorted By Rating



Filter Games by Platform



Get Highest Rated Game



Coding

The project is implemented in Python using the following features:

- **Modules Used:** csv, os
- **File Handling:** All records are stored in a CSV file (game_info_system.csv) using the csv module.
- **Functions Implemented:**
  - add_game() – Adds a new games.
  - display_games() – Displays all games from the file.
  - delete_game() – Deletes a games by their name.
  - update_game() – Updates individual fields of a specific games.
  - search_game() – Searches for a particular game by name.
  - sorted_by_rating() – Sort game by their rating.
  - Filter_game_by_platform() – filter game by platform.
  - initialize_file() – Ensures the CSV file is initialized with headers.

The program follows a **modular structure**, making the code maintainable and extendable.

CONCLUSION

- The **Game Info Management System** is a simple yet powerful application designed to efficiently manage and organize game-related data. It provides a user-friendly interface for adding, viewing, updating, deleting, and searching game records stored in a CSV file. With features like filtering by platform, sorting by rating, and identifying the highest-rated game, the system proves to be a valuable tool for users who want to keep track of various games and their details.

- Throughout the development of this project, we have deepened our understanding of Python's capabilities in **file handling**, **data validation**, **conditional logic**, and **user-centric design**. The project demonstrates how a cleanly structured, menu-driven application can provide significant practical utility while maintaining simplicity. Future enhancements could include adding **graphical interfaces (GUI)** for improved user interaction, incorporating **analytics** for P&L summaries, or integrating with **real-time APIs** for dynamic data entry and insights. Transitioning from CSV files to a **relational database system** can also be considered for better scalability and performance.

LEARNING DURING SIP

During the course of the Summer Internship Project (MINI PROJECT), I gained significant technical and practical knowledge, both in programming and in understanding the financial markets. The key areas of learning are summarized below:

Technical Learnings:

- **Python Programming:**
  - Gained hands-on experience in writing modular, structured, and readable Python code.
  - Learned to implement core programming concepts such as conditional statements, loops, user input handling, and function-based program structure.

- **CSV File Handling:**
  - Understood how to read from, write to, and update records in CSV files using Python's csv module.
  - Learned to manipulate file data securely, ensuring game record integrity and uniqueness.

- **CLI-Based User Interface:**
  - Designed a text-based interface that simplifies the user's interaction with the system.
  - Ensured smooth user experience through a menu-driven approach.

Project Development Skills:

- **System Design:**
  - Applied structured design techniques like DFDs and flowcharts to visualize system workflow before development.

- **Testing & Debugging:**
  - Practiced iterative testing and debugging to ensure accurate and efficient operation of all features.

- **Documentation:**
  - Learned how to document a project thoroughly, including synopsis preparation, design diagrams, and user-oriented documentation.

BIBLIOGRAPHY

Online References

1. W3Schools – Python File Handling
   Referred for file operations and input/output techniques.

2. GeeksforGeeks – Python Projects
   Used for getting inspiration and coding patterns for CLI-based Python applications.

3. https://realpython.com/python-csv/
   Real Python – CSV File Operations

Offline References

1. College lecture handouts for Python programming and CLI development

2. MINI PROJECT journal provided by faculty for system design diagrams and documentation tips