Forestview Technologies

Talha Ali, Amran Rahim, Soham Bhavsar, Maher Harkati, Mohammed Hoque

## Why Our Team Should Be Chosen:

* Our team at Forestview Technologies is highly suited for this new venture because we prioritize efficiency and innovation. We recently embarked on creating an arcade-style video game platform, featuring a variety of games. Initially, we identified the core requirements for the arcade system, focusing on user-friendliness and accessibility. We adopted a development approach akin to the waterfall model but enhanced task division more thoroughly than in our previous projects.
* We began development by constructing an interface that resembles a classic arcade, ensuring that the user experience was intuitive and straightforward, enabling players to navigate between games effortlessly. We committed to streamlining the process, ensuring that even the most complex operations were manageable within a few interactions.
* Our team divided the workload effectively, with dedicated groups focusing on different aspects of the platform—from game development to the user interface. After constructing the system, we moved into a rigorous testing phase, identifying and resolving any issues to refine the gameplay and overall functionality.
* Upon completing the testing, we launched the arcade platform, performed a comprehensive review, and executed final adjustments. This project, like our previous endeavors, was methodically structured step-by-step under the waterfall model, which significantly aided in maintaining clarity and producing a high-quality arcade game platform. Our adeptness in choosing and implementing the right development strategy is why you should consider us for this project.

## Tools We Used:

**Integrated Development Environment (IDE):**

We relied on Visual Studio Code (VS Code) as our primary IDE for coding tasks. VS Code provided a robust set of features, including syntax highlighting, IntelliSense code completion, and Git integration, which streamlined our development workflow. Its extensive library of extensions allowed us to customize our environment according to project requirements.

**Backend and Frontend Development:**

For backend logic and frontend presentation, we employed Python for its versatility and rich standard library. Python facilitated rapid development and implementation of core functionalities. Additionally, HTML & CSS were utilized for structuring web content and styling user interfaces, with CSS frameworks like Bootstrap 5 ensuring responsive designs across various devices.

**Version Control and Collaboration:**

GitHub served as our centralized version control platform, enabling seamless collaboration, code sharing, and project management. We leveraged features such as repositories, forks, pull requests, and issue tracking to coordinate development efforts effectively. Moreover, GitHub Desktop provided a graphical interface for version control tasks, simplifying branching, committing changes, and syncing with remote repositories.

**Static Code Analysis:**

To ensure code quality and adherence to coding standards, we utilized Pylint for static code analysis. Pylint's comprehensive checks and customizable configurations helped maintain consistency, readability, and robustness throughout our codebase, contributing to overall software quality.

**Cloud Storage and File Sharing:**

OneDrive served as our cloud storage solution, facilitating secure storage, sharing, and access to project files and resources. Its seamless integration with desktop and web applications, along with real-time collaboration features, enhanced team productivity and ensured data accessibility from anywhere.

**Executable File Generation:**

For deployment and distribution of our application, we employed PyInstaller to convert Python scripts into standalone executable files (.exe). PyInstaller's bundling of the Python interpreter and dependencies simplified usage for end-users, eliminating the need for separate installations.

**Data Serialization:**

We utilized JSON (JavaScript Object Notation) for serializing and deserializing data structures, such as storing inventory information and saving orders. JSON's lightweight and human-readable format provided an efficient means of data storage and interchange between different systems.

## Desired Need & Fulfillment:

* Forestview Technologies is the optimal choice to fulfill the desired needs outlined in the contract. The team's commitment to efficiency is exemplified by the successful completion of the project ahead of schedule, ensuring both cost-effectiveness and timely delivery. Through a meticulous analysis of requirements, we developed clear guidelines and a system design, aligning the website theme with the software's functionality to meet the client's vision. In response to the desired need for an innovative application, our team crafted a wholesaling application for product inventory, showcasing adaptability and creative problem-solving. The implementation phase prioritized achieving optimal time complexity, with each functionality fine-tuned to ensure efficiency. Rigorous testing, a seamless deployment process, and detailed maintenance, including final debugging and additional functionalities, demonstrated our commitment to delivering a reliable product. Our strategic use of the waterfall model in the software development cycle, coupled with an adaptive approach to choosing methodologies based on client needs, underscores our ability to provide a structured and client-tailored solution. Utilizing a suite of tools, including VS Code, Python with Tkinter, HTML, CSS, JavaScript, OneDrive, Pylint, and GitHub, facilitated effective collaboration and version control. In conclusion, Forestview Technologies not only meets but surpasses the desired needs outlined in the contract, offering a comprehensive and detail-oriented approach to ensure the project's success.

## Team Collaboration:

Within the collaborative framework of Forestview Technologies, each team member brings unique expertise, contributing to the project's success. For this project, our team had all its members focus primarily on the application coding. We split the team between functionality and UI for both the customer and manager experiences.

Talha Ali and Soham Bhavsar – Core Functionality for Manager Experience:

- Talha Ali and Soham Bhavsar have played instrumental roles in the application coding phase, leveraging their programming skills and creativity to develop a robust and innovative inventory management system. They created the framework for a rudimentary password system, an automatic inventory system to show current stock, a daily time sheet for employee shifts, a revenue tracker, and finally a search function to search orders using customer name. All data was stored in a .JSON file for easy exporting and future use.

Amran Rahim – UI for Manager Experience and Coding Compliance:

- Amran Rahim used the tkinter python library to create a robust and accessible UI for our manager experience. When the button to switch to it is pressed, the UI first prompts the user for the admin password. Once the password is entered, the user is able to move on to the main manager screen. Amran created an easy to parse UI to display all the relevant data in text boxes, as well as additional buttons to search orders, add/remove employees, and switch back to the pizza ordering system. Amran Rahim also took on the crucial responsibility of coding compliance, utilizing tools like Pylint to conduct code analysis. His attention to detail ensured that the code adhered to industry standards, minimizing errors, and enhancing the overall quality of the project.

Maher Harkati - Core Functionality and Frontend Development:

- Maher Harkati created the core framework and logic for the application coding of the customer experience. In addition, he added a ticketing system to one of the games so the user may cashout the tickets. Maher Harkati also took charge of the front-end development, updating the website with all the project 3 information and links. His contributions extended to logic implementation using JavaScript, ensuring a seamless and engaging user experience.

Mohammed Hoque – UI and Core functionality:

* Mohammed Hoque focused on using tkinter to create the UI and Pygame library for the user to choose between games. He created a colorful menu with easy options for the arcade type of games to choose from which are Letter Guessing Game, Dice Guessing Game, Simon Game. He also has fields for the customer's name, phone number, and order quantities. Lastly, he added two buttons: One to place the order, and another to switch the inventory management system.

Together, the collaborative efforts of Talha, Soham, Maher, Mohammed, and Amran synergized seamlessly, showcasing the effectiveness of their teamwork. The team's collective skills in application coding, frontend development, and coding compliance harmonized to create a successful and comprehensive solution for the client.

## Hosting Quotes:

## 

## In exploring hosting solutions for our Arcade application website, the pricing considerations vary among different providers. Amazon Web Services (AWS) offers plans ranging from $15 - $40 per month, with the flexibility to scale based on traffic. Microsoft Azure provides options starting at $20 per month, featuring robust cloud services tailored to diverse needs. On the other hand, Bluehost, a user-friendly platform, starts at $8 per month, making it a budget-friendly alternative with solid performance.

## Future Recommendations:

## To enhance Forestview Technologies' Arcade Game application, several key recommendations can be implemented. We can Prioritize scalability through flexible cloud solutions, enhance security measures for data protection, and explore mobile responsiveness. Seamless integration with third-party services. Improving the user interface with visual enhancements and a more organized layout will enhance user experience and navigation. Strengthening error handling and validation mechanisms ensures users receive informative feedback and input accuracy. Offering Multiple games that allowed the users to pick one they would perfer. Implementing order management features like tracking and modification capabilities boosts user convenience. Enhancements to inventory management, such as real-time updates and automated reordering, ensure smooth operations. Reporting and analytics provide valuable insights for strategic planning. Localization, performance optimization, security measures, and integration with external systems round out the enhancements, delivering a robust, efficient, and user-friendly pizza ordering platform for Forestview Technologies.

## STEM principle applications:

The arcade UI presented here showcases various games employing STEM principles, offering entertainment while subtly engaging users in scientific and mathematical concepts. One notable application is the "Click Game," where players must click on randomly appearing objects, demonstrating principles of probability and reaction time. The unpredictability of the object's appearance mimics random events in nature, while the player's response time reflects human reflexes, both concepts fundamental to STEM fields like statistics and physiology.

Furthermore, the "Letter Guessing Game" and "Dice Guessing Game" incorporate elements of probability and decision-making. Players must make educated guesses based on available information, akin to hypothesis testing or decision-making processes in scientific research. These games offer an interactive way to understand and apply statistical concepts in a fun and engaging manner.

Additionally, the "Simon Game" challenges players to remember and replicate sequences, exercising cognitive skills such as memory and pattern recognition. This mirrors tasks encountered in computer science, where algorithms often rely on pattern recognition and sequence processing, fostering computational thinking and problem-solving abilities.

Overall, through these diverse games, the arcade UI not only provides entertainment but also promotes STEM education by integrating key principles and concepts into interactive gameplay experiences.

## Technical Documentation:

### Code Overview

The Arcade UI is a graphical user interface (GUI) application developed using Python's Tkinter library. It offers a collection of interactive games for users to enjoy. The application features various games, including a click game, letter guessing game, dice guessing game, and Simon game.

### Features-

#### Click Game:

The Click Game presents users with a delightful opportunity to interact with objects on the screen. Key features include:

* Object Interaction: Users can click on randomly generated objects, distinguished by their vibrant blue or red colors.
* Score and Tickets: Clicking on blue objects increments the player's score and ticket count, fostering a sense of achievement. Conversely, clicking on red objects deducts points.
* Sound Effects: The game comes alive with immersive sound effects, providing auditory cues for each interaction.

Expansion Ideas:

* Leaderboard: Introduce a leaderboard to showcase top scores achieved by players, fostering healthy competition.
* Power-Ups: Inject excitement by incorporating power-ups that offer temporary advantages or bonus points.

#### Letter Guessing Game:

In the Letter Guessing Game, users are tasked with guessing a letter from a predetermined range. Key features include:

* User Engagement: Users engage by guessing letters within a predefined range, receiving immediate feedback on their attempts.
* Scoring Mechanism: Each correct guess boosts the player's score, encouraging persistence and strategic thinking.
* Sound Feedback: Sound effects accompany every guess, enriching the gaming experience.

Expansion Ideas:

* Difficulty Levels: Enhance replayability by implementing varying difficulty levels, catering to players of different skill levels.
* Multiplayer Mode: Foster social interaction with a multiplayer mode, allowing users to compete in real-time letter guessing challenges

#### Dice Guessing Game:

The Dice Guessing Game invites users to predict the outcome of a virtual dice roll. Key features include:

* User Engagement: Users actively participate by guessing numbers between 1 and 6, receiving feedback on their accuracy.
* Reward System: Successful guesses contribute to the player's score, fostering a sense of accomplishment.
* Sound Effects: Engaging sound effects accompany each guess, heightening the overall gaming experience.

Expansion Ideas:

* Customizable Range: Empower users with the ability to customize the range of numbers for an added layer of personalization.
* Streak Tracking: Implement streak tracking to reward consecutive correct guesses, encouraging sustained engagement.

#### Simon Game:

The Simon Game challenges users to replicate a sequence of numbers or letters, testing their memory prowess. Key features include:

* Memory Challenge: Users exercise their memory by recalling and reproducing sequences presented by the game.
* Validation: The game verifies user input against the generated sequence, providing instant feedback on correctness.
* Immersive Experience: Sound effects enrich the gameplay, enhancing user immersion and enjoyment.

Expansion Ideas:

* Advanced Challenges: Introduce advanced challenges with longer sequences or mixed patterns to elevate the gameplay complexity.
* Timed Mode: Heighten the stakes with a timed mode, challenging players to complete sequences within a specified timeframe.

### User Interaction:

Game Instructions:

Each game interface provides comprehensive instructions, elucidating rules, controls, and scoring mechanisms, ensuring users can engage effortlessly.

Visual Enhancements:

Dynamic visual effects and animations captivate users, augmenting the gaming experience and immersing them in a visually stimulating environment.

### Usage

**Playing Games:**

Click on the respective buttons to start playing each game.

Follow the instructions provided within each game interface.

**Cashing Out Tickets:**

Click on the "Cash Out Tickets" button to see how many tickets you've earned.

Resets the ticket count after cashing out.

**Dependencies:**

**Tkinter Library:**

* GUI Creation: Describe how the Tkinter library is used to create the graphical user interface (GUI) of the Arcade UI application. It provides essential components such as buttons, labels, and frames for building interactive elements.
* Cross-Platform Compatibility: Highlight Tkinter's cross-platform compatibility, ensuring that the Arcade UI can be used on various operating systems, including Windows, macOS, and Linux.
* Ease of Use: Emphasize Tkinter's simplicity and ease of use, making it accessible to both novice and experienced developers for designing intuitive user interfaces.

**Pygame Library:**

* Sound Effects: Explain how the pygame library is utilized to incorporate sound effects into the Arcade UI application. It enables the playback of audio files to enhance the gaming experience.
* Audio Management: Describe pygame's capabilities for managing audio resources, including loading, playing, and controlling sound effects, ensuring smooth integration with the user interface and gameplay.
* Performance Optimization: Mention pygame's efficiency in handling multimedia tasks, such as audio playback, without compromising performance, resulting in a seamless gaming experience for users.

### **Conclusion**

The Arcade UI provides an entertaining platform for users to enjoy various interactive games. With its user-friendly interface and diverse game selection, it offers a fun experience for users of all ages. The incorporation of sound effects enhances the gaming experience, adding excitement to each interaction.