Forestview Technologies

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

## Why Our Team Should Be Chosen:

* Our team at Forestview Technologies is great for this new project because we work fast and smart. We're making a pizza parlor simulator with a special section for managers to schedule work times and add employees. First, we figured out what the pizza parlor needed. We decided to make a system that looks and feels like a real pizza shop, where managers can easily handle staff schedules and details. Then, we started building it. We wanted everything to work quickly and without unnecessary steps, so we made sure even complex tasks could be done with a few clicks. We got straight to programming, keeping things simple to ensure the system was fast and reliable. We split up the work, focusing on different parts like the customer area and the admin side for employees and schedules. After building it, we tested everything to catch any mistakes. We made sure that the part where employees get added and schedules get made was working perfectly with the rest of the simulation. Finally, we launched it, checked it all over again, and made some last-minute fixes. This project, just like our previous ones, followed the waterfall model. This means we did everything step by step, which made it easier to keep everything organized and make a great pizza parlor simulation. Our skill in picking the right way to do the project is why you should pick us for the job.

## 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 success of the project. 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 for Customer Experience and Frontend Development:

- Maher Harkati created the core framework and logic for the application coding of the customer experience. He created a database for the ingredients and pricing data in the form of a .JSON. He then created a system for customers to order pizza by selecting the type, size, crust, and additional toppings. Maher then stores the order is then stored under a name and number. 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 for Customer Experience:

* Mohammed Hoque focused on using tkinter to create the UI for the customer ordering experience. He created a colorful menu with a simple pizza image, and easy to select options for the pizza type, size, crust, and toppings. 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 Pizza ordering 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' pizza ordering system, 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 more customization options like additional toppings and personalized pizza configurations adds value. 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:

Our project seamlessly integrated mathematical and scientific principles across various stages, showcasing the versatility of STEM disciplines in software development:

In backend development using Python, mathematical logic and algorithms were instrumental in crafting efficient solutions. From calculating inventory quantities to optimizing data structures for processing orders, mathematical operations and algorithmic strategies were applied to ensure precision and performance in our software.

Handling data with JSON serialization and storage solutions provided opportunities for data analysis and visualization. Leveraging Python libraries like Pandas and Matplotlib, we delved into data trends, patterns, and correlations. These insights guided decision-making and facilitated process improvements, showcasing the power of data-driven approaches in software development.

Frontend design using HTML, CSS, and Bootstrap incorporated principles of geometry and layout design. By leveraging concepts such as grid systems, typography, and color theory, we crafted visually appealing and user-friendly interfaces. Our responsive designs seamlessly adapt to diverse screen sizes and resolutions, enhancing the user experience.

Throughout the development process, computational thinking guided our problem-solving approach. Decomposing complex tasks into manageable subproblems, identifying patterns, and developing step-by-step solutions were fundamental strategies. This approach enabled us to effectively tackle coding challenges and create robust software solutions.

Analyzing user feedback and testing data involved employing statistical methods and probabilistic reasoning. Techniques like hypothesis testing and regression analysis provided valuable insights into user preferences and areas for improvement. These statistical approaches validated the effectiveness of our software solutions, ensuring they met user needs and expectations.

Overall, our project exemplifies the interdisciplinary nature of STEM principles in software development. By integrating mathematical, scientific, and computational thinking, we created innovative solutions that address real-world challenges and enhance user experiences.

## Technical Documentation:

## **Overview**

## The Pizza Ordering and Inventory Management System is a graphical user interface (GUI) application developed using Python's Tkinter library. The system facilitates both pizza ordering functionality for customers and inventory management for administrators. It allows users to place pizza orders, manage inventory, track orders, and monitor revenue. The system comprises two main components: the Pizza Ordering System and the Inventory Management System.

## **Features**

1. **Pizza Ordering System**:
   * Allows customers to place pizza orders with ease.
   * Provides a graphical interface for selecting pizza type, size, crust, toppings, and quantity.
   * Validates user inputs to ensure accurate orders.
   * Checks inventory levels to guarantee sufficient ingredients for each order.
   * Deducts the required ingredients from the inventory upon successful order placement.
   * Records order details, including timestamp, for future reference.
   * Provides feedback to the user regarding the status of the order.
2. **Inventory Management System**:
   * Enables administrators to manage pizza inventory and track revenue.
   * Requires authentication with an administrator password for access.
   * Displays current inventory status, including ingredient quantities for each pizza type.
   * Allows administrators to update inventory levels based on ingredient usage.
   * Tracks and displays total revenue generated from pizza orders.
   * Provides seamless switching between inventory management and pizza ordering functionalities.
3. **Timesheet System**:
   * Assists administrators in managing employee schedules efficiently.
   * Offers a GUI interface for adding and removing employees from specific timeslots.
   * Helps in organizing employee shifts and optimizing staffing levels based on demand.
   * Integrates seamlessly with the overall system architecture, providing a comprehensive solution for restaurant management.

## **Code Structure**

### **Functions**

**Ordering System:**

* **Order Placement Method (def place\_order(self):)**:
  + Handles the process of placing an order for pizza.
  + Retrieves user-selected options such as pizza type, size, crust, quantity, and customer details.
  + Validates user inputs, checks inventory availability, and processes the order accordingly.
  + Displays appropriate error messages or confirmation dialogs based on the outcome of the order placement.
* **GUI Creation Method (def create\_order\_gui(self):)**:
  + Defines a method responsible for creating the ordering GUI components.
  + Constructs the graphical user interface for users to select pizza options, specify quantity, and enter customer details.
  + Utilizes Tkinter widgets such as labels, dropdown menus, entry fields, and buttons to design the ordering interface.
* **Inventory Check (for topping, amount in self.inventory[pizza].items():)**:
  + Verifies whether the inventory has sufficient ingredients to fulfill the order.
  + Iterates through the ingredients required for the selected pizza and checks if the available quantity meets the requested amount.
  + Ensures that the order can be processed without running out of stock for any ingredient.

**Inventory Management:**

* **Password Authentication Method (def check\_password(self):)**:
  + Verifies the administrator's password to grant access to the inventory management system.
  + Retrieves the entered password from the GUI entry field and compares it with the predefined admin password.
  + Allows access to the management GUI only if the correct password is provided.
* **Inventory Display (self.inventory\_listbox = tk.Listbox(self.root, width=50, height=15), self.update\_inventory\_list())**:
  + Creates a graphical representation of the current inventory items and their quantities.
  + Utilizes a Tkinter Listbox widget to display the inventory data in a scrollable list format within the management GUI.
  + Updates the inventory list dynamically to reflect any changes made to the inventory (e.g., ingredient deductions after order placement).
* **Revenue Calculation (def calculate\_revenue(self):)**:
  + Computes the total revenue generated from pizza orders.
  + Iterates through the list of orders and calculates the total revenue by multiplying the quantity of each pizza order by its price (assuming a fixed price per pizza).
  + Updates the revenue label in the management GUI to display the current revenue amount.

## **Usage**

1. **Ordering Pizza**:
   * Select pizza, size, crust, and toppings.
   * Enter customer information.
   * Click "Place Order".
2. **Managing Inventory**:
   * Enter admin password.
   * View inventory and revenue.
   * Add/remove employees from timeslots.
   * Search orders by customer name.
3. **Switching Systems**:
   * Use buttons to switch between ordering and management.
4. **Exit**:
   * Close the program window.

## **Dependencies** The program relies on the following:

* Tkinter library for creating the GUI.
* **json**: The **json** module in Python provides functions to work with JSON data. In this code, it is used to load and save orders to a JSON file (**orders.json**).
* **datetime**: The **datetime** module supplies classes for manipulating dates and times in Python.

## **Conclusion**

The Pizza Ordering and Inventory Management System provides a comprehensive solution for managing pizza orders, inventory, revenue, and employee schedules within a single application. Its user-friendly interface and robust functionality make it suitable for use in pizzerias and restaurants looking to streamline their operations and enhance customer satisfaction. The modular design allows for easy scalability and customization to meet the specific needs of different establishments.