# **CHAPTER SIX**

## **6.0 SYSTEM IMPLEMENTATION**

### **6.1 Tools used for coding**

To develop and test the website described above, the following tools can be utilized:

**Front-end Development**

* HTML: Markup language for creating web pages.
* CSS: Styling language for designing the website's appearance.
* JavaScript: Programming language for implementing interactive features.

**Back-end Development**

* Django Framework: Python-based web framework for building robust and scalable web applications.
* Django ORM: Object-Relational Mapping tool for interacting with the database.
* MySQL: Databases supported by Django for storing user data and order information.

**Development Environment**

* Text Editors: Visual Studio Code, Sublime Text, or Atom for coding.
* Version Control: Git for managing source code changes.
* Local Development Server: Django's built-in development server for running the website during development.
* Browser Developer Tools: Chrome DevTools and Firefox Developer Tools for debugging and testing.

### **6.2 Testing**

Testing the website can involve various types of testing, including:

**Functional Testing**

* Validate that all links on the home page navigate to the correct pages.
* Verify that the signup and login functionalities work as expected.
* Test the customer dashboard to ensure it displays the available restaurants and other links accurately.
* Check the restaurant orders page to confirm that it shows the correct orders received.

**Data Testing**

* Test signup and login forms with valid and invalid inputs.
* Verify that user data is correctly stored and retrieved from the database using Django's ORM.
* Test the menu page to ensure it displays the correct restaurant menu items.

**Usability Testing**

* Evaluate the user experience of navigating the website and performing tasks.
* Check if the layout and design are visually appealing and intuitive.
* Test the responsiveness of the website on different devices and screen sizes.

**Payment Testing**

* Test the payment process using different payment statuses,
* Verify that successful payments redirect the user to the order confirmation page.
* Test failed payment scenarios to ensure users are redirected back to the customer dashboard.

### **6.3 Proposed Change-over Techniques**

When implementing changes or updates to the website, the following change-over techniques can be considered:

**Incremental Change-over**

* Implement updates in small increments rather than making extensive changes all at once.
* Test each increment thoroughly before deploying it to the production environment.
* This approach allows for easier identification and resolution of issues.

**A/B Testing**

* Implement changes on a subset of users, while another group continues using the current version.
* Collect feedback and compare metrics (e.g., conversion rates, user satisfaction) between the two groups.
* Gradually roll out the changes based on the results obtained.

**Staged Rollout**

* Deploy changes to a limited set of users or specific geographical regions.
* Monitor the system closely for any issues or performance bottlenecks.
* Gradually increase the rollout to a larger user base once stability is ensured.

**Canary Release**

* Introduce new features or updates to a small percentage of users initially.
* Monitor the system's performance and gather user feedback.
* Based on the feedback and performance metrics, decide whether to proceed with the full release or make necessary adjustments.

By utilizing Django framework for back-end development, conducting comprehensive testing, and implementing effective change-over techniques, the Online Customizable food ordering system can be developed, maintained, and updated successfully, providing a seamless and personalized food ordering experience to users.