Online Grocery Order System

Detailed Project Report (DPR)

1. **Executive Summary**
    - *The Online Grocery Order System project was initiated to meet the increasing demand for online grocery shopping. This web-based application was designed to provide customers with a convenient platform to order groceries online, streamlining the entire process from product selection to delivery address submission.*
2. **Introduction**
    - **Background:**
        - The Online Grocery Order System project was initiated in response to the growing demand for convenient online grocery shopping.
    - **Motivation:**
        - The motivation behind the project was to provide users with a user-friendly platform to order groceries from the comfort of their homes.
    - **Project Goals:**
        - The primary goal of this project was to develop and deploy a functional online grocery ordering system to streamline the grocery shopping experience for customers.

3. **Project Scope**
    - **Inclusions and Exclusions:**
        - The project scope includes the development of a web and mobile application for customers to order groceries online.
    - **Limitations and Constraints:**
        - The project was limited to a predefined local area for initial testing.
4. **Project Planning and Management**
    - **Project Management Approach:**
        - Agile methodology was adopted for project management.
    - **Project Schedule:**
        - The project schedule was created with milestones including design, development, testing, and deployment phases.
    - **Resource Allocation:**
        - The project team consisted of two students
    - **Risk Assessment:**
        - Currently under analysis and updates to the web app would be provided in future
5. **System Overview**
    - **Description:**
        - The Online Grocery Order System is a comprehensive platform that allows customers to browse, select, and order groceries online.

- **Architecture:**
  - The system follows a client-server architecture with a web client interacting with server-side components.
- **Key Features:**
  - Features include product catalog browsing, shopping cart management, order placement, and notifications.

6. **Functional Requirements**
   - **User Stories:**
     - User stories were used to define the user interactions with the system.
   - **Detailed Requirements:**
     - Detailed functional requirements were documented, including user account creation, product catalog management, and order processing.
7. **Non-functional Requirements**
   - **Performance Requirements:**
     - Performance benchmarks included response times and system uptime.
   - **Security Measures:**
     - Security measures included authentication, encryption, and secure payment processing.
   - **Usability and Accessibility:**
     - The system was designed for ease of use and accessibility for all users.
   - **Availability and Scalability:**
     - High availability was a goal, with scalability for increased user loads.

8. **System Design**
    - **System Architecture:**
        - Diagrams illustrated the system's architecture with frontend and backend components.
    - **Database Schema:**
        - The database schema included tables for user data, products, orders, and notifications.
    - **User Interface Design:**
        - Wireframes and mockups demonstrated the user interface.
9. **Implementation**
    - **Technology Stack:**
        - Technologies included HTML, CSS, TypeScript, React, ASP.NET, SQL, Microsoft Azure, Swagger API
    - **Development Process:**
        - The Agile development process included iterative development cycles.
    - **Challenges and Solutions:**
        - Challenges included integrating server backend setup and payment gateways
10. **Testing and Quality Assurance**
    - **Test Plan:**
        - The test plan outlined test cases, scenarios, and expected outcomes.
    - **Testing Types:**
        - Testing types included unit testing, integration testing, system testing, and user acceptance testing.

- **Test Results:**
  - Test results were documented, including issues found and resolutions.

11. **Deployment**
    - **Deployment Strategy:**
      - The system was deployed in a staged environment before production.
    - **Environment Setup:**
      - Server setup included configuring web servers, application servers, and databases.
    - **Deployment Process:**
      - The deployment process was documented step-by-step.
    - **Post-Deployment Testing:**
      - Post-deployment testing ensured a successful rollout.
12. **Performance Evaluation**
    - **Performance Testing:**
      - Performance testing results included response times and resource utilization.
    - **Scalability Assessment:**
      - Scalability was tested to accommodate increased users.
    - **Optimization Measures:**
      - Optimization measures included database indexing and caching.
13. **Security Assessment**
    - **Security Testing:**

        - Security testing involved vulnerability scanning and penetration testing. Azure Wins.
    - **Security Findings:**
        - Findings were addressed, and security enhancements were implemented. Azure Wins.
14. **User Documentation**
    - **User Manuals:**
        - User manuals and guides were provided for customers.
    - **Admin Documentation:**
        - Documentation for administrators covered system management.
15. **Maintenance and Support**
    - **Maintenance Plan:**
        - A maintenance plan included regular updates and bug fixes.
    - **Bug Reporting:**
        - A bug reporting system allowed users to report issues.
    - **Web Updates:**
        - UI updates and improvements were planned for feature enhancements and security patches.
    - **User Support:**
        - User support mechanisms included email and a helpdesk.
16. **Conclusion**
    - **Summary:**
        - The project achieved its goals, resulting in a functional Online Grocery Order System.
    - **Lessons Learned:**

        - Lessons learned included the importance of thorough testing and user feedback.
    - **Future Enhancements:**
        - Future enhancements could include web app improvements, geographic expansion, more payment gateway options
17. **References**
    - *Coursera React Full Stack Workshop Course, SQL Guide, Visual Studio Guide, .NET Mastery Course, API From Scratch Guide*
