PlacementPal : Graduation and Placement Predictor

Design Specification

1. System Architecture

- ➔ The server will be housed on a cloud platform for scalability and stability.
- ➔ The system will be created using a client-server architecture.
- ➔ A RESTful API will be used by the system for client-server communication.
- ➔ Client applications for mobile platforms (Android and iOS) and web browsers will be developed.

2. User Interface Design

- Simple and contemporary user interface will be used.
- It will be adaptable to different screen sizes and devices because to its responsiveness.
- The same branding and color schemes will be used across all platforms and pages.
- The user experience will be improved through simple controls and navigation.

3. Database Design

The system will make use of a relational database management system (RDBMS).
The database schema includes tables for users, products, orders, and inventory.
Data will be standardized to decrease redundancy and boost data integrity.
Appropriate indexing and query optimization must be used for effective data retrieval.

4. Security Design

Industry-standard encryption methods will be used to carry out user authentication.
Role-based access control and permission will limit access to sensitive functionalities.
Data encryption will be applied to secure user data and transactions.
Regular security audits and vulnerability analyses will be carried out.

5. Functionality and Features

User Account Creation and Secure Login: Users can create accounts and safely log in.
Users can browse and search the Product Catalog, view product details, and add goods to their shopping cart.
Users can securely manage their shopping carts and complete purchases using the shopping cart and checkout features.

Users can view order statuses and order histories in Order Management.
Inventory Management: Administrators have the power to control product inventory and make changes to product information.
Reporting and analytics: Produce sales reports and assess user activity.
Users receive email or push alerts with order confirmations and status updates.

6. **Performance Considerations**

The system will be configured to have the lowest latency and highest performance possible.
Caching techniques will be used to cut down on database requests.
We'll employ load balancing to distribute traffic equally.
Scalability will be attained by deploying extra server resources as necessary.

7. **Testing and Quality Assurance**

Unit testing, integration testing, and user acceptability testing will all be conducted.

There will be the creation of test cases and test scripts to confirm functionality and dependability.

Performance evaluations will validate the system's scalability and responsiveness.

Security testing will reveal vulnerabilities that will be fixed.

8. **Documentation**

Complete technical documentation will be created.
User manuals will be provided to administrators and end users.