* **Functional & non-functional requirement**

|  |  |
| --- | --- |
| * **Functional requirements** | |
| **User Registration and Authentication** | * Users should be able to register, create accounts, and log in securely. * Implement password recovery and account verification processes. |
| **Order Creation** | * Allow users to create new orders. * Create a simple request form with fields for entering basic information such as the name of the desired item, the quantity needed, available budget, specific specifications for the requested item, and any other relevant information. * We consider additional options, such as attaching images or files to provide more information about the requested item. |
| **Location and Address Selection** | * Provide an interface for users to input or select their delivery location and address. * Include options for saving multiple delivery addresses. |
| **Order Confirmation and Payment** | * Confirm the order details, including the total cost, and prompt users for payment. * Support various payment methods, such as credit cards, digital wallets, or cash on delivery. * The user needs a “Submit Request” button to send the information they have entered. |
| **Display Requests and Interaction** | * Display user requests to service providers who can respond to them. * Providing an interface for service providers to interact with requests by presenting their offers. * Enable service providers to submit their offers and specify prices and terms. |
| **User Selection** **of Preferred Offer** | * Grant users the ability to choose from among the service providers based on the offers and terms presented. * Add a "Select" or "Accept" button for users to choose from among the offers. * Update the status of the request and notify the user of their selection. |
| **Request Processing and Assignment** | * Implement a system to process requests and assign them to selected service providers. * Provide options for service providers to interact with posts by using "Accept" or "Offer" buttons. * Develop a dashboard for managing offers submitted by service providers, including specifying prices and terms. |
| **Notifications and Alerts** | * Implement a real-time notification system to inform users of offer details and request status. * Send notifications to both users and service providers regarding request status. |
| **Request Tracking and Delivery** | * Allow users to track the status of their requests and receive updates on the delivery process. * Send automatic updates on the provider's location and estimated arrival time. |
| **User Ratings and Comments** | * Enable users to provide ratings and comments after the service or delivery is completed. * Display the average ratings for users. |
| **Customer Support** | * Develop a customer support system that allows users and service providers to communicate and resolve issues. |
| **Admin Dashboard** | * Create an administrative control panel that allows administrators to manage posts, offers, and monitor operations. |
| * **Non-functional requirements** | |
| **Security** | * The website must implement high levels of security to protect sensitive information and prevent security breaches and protect user data from unauthorized access. * User authentication and identity verification mechanisms must be secure. |
| **performance** | * + The website should load quickly and provide good performance to ensure a smooth user experience,   even with a large number of users.   * + Performance can be improved by reducing loading times, optimizing images, and utilizing caching.   + The system supports a large number of concurrent users at all other times. |
| **Availability** | * The website must be available 24/7/365 and maintain a high level of uptime. * Contingency plans and disaster recovery strategies should be in place in case of website failure. |
| **Cross-Browser Compatibility** | * The website should be designed to work consistently across different common web browsers (Chrome, Firefox, Safari, etc.). |
| **Responsive Design** | * The website should be responsive, adapting to various devices and screen sizes. |
| **Ease of Maintenance** | * Code should be well-organized and maintainable to facilitate future maintenance and development. |
| **Scalability** | * The website must be continuously scalable to accommodate the increase in the number of users and the volume of data. |
| **Database Performance** | * The database should be well-structured and optimized for fast response times. |
| **Documentation** | * The project and codebase should be well-documented to ease future handling by development teams. |
| **Data Security** | * Personal data and sensitive information must be protected with strict information security measures. |
| **Continuous Improvement** | * There should be a plan for ongoing management and implementation of improvements to the website. |
| **Compliance with Standards** | * The website must comply with international web standards such as W3C and WCAG for accessibility. |
| **Usability** | * The system must be easy to use and understand. |