User Roles for Project

Front End Web Designer - Mohebul Hasan Emon

Key Responsibilities

- UI/UX Design:
    - Design the overall look and feel of the website.
    - Create wireframes, prototypes, and mock-ups to showcase layout and interactions.
    - Focus on enhancing user experience (UX) by making the interface intuitive, accessible and user-friendly.

- HTML/CSS:
    - Write clean, semantic HTML for the structure of the web page.
    - Style the page using CSS to ensure it looks attractive and is responsive to access different screen sizes.

Tools & Technologies:

- **Frontend Libraries & Frameworks:** React, Vue.js, Angular, Bootstrap.
- **Preprocessors & Tools:** Sass, LESS, Gulp, Webpack.
- **Version Control Systems:** Git, GitHub, GitLab, Bitbucket.
- **Testing & Debugging Tools:** Chrome DevTools, BrowserStack.

Backend Designer: - Rajababu Kushwaha

Key Responsibilities:

- **Server-Side Logic:**
    - Develop the application’s back-end logic, which handles requests from the front end (e.g., processing form submissions and user authentication).
    - Work with back-end frameworks (e.g., Node.js, Django, Ruby on Rails) to define the application's behaviour.

- **Database Management:**
    - Design and manage databases (e.g., MySQL, PostgreSQL, MongoDB).
    - Implement CRUD (Create, Read, Update, Delete) operations and ensure the system handles data effectively.
    - Use **ORMs (Object-Relational Mappers)** such as Sequelize (Node.js) or Django ORM (Python) to simplify database interactions.

- **API Development:**
    - Design and build **RESTful** or **GraphQL APIs** to allow communication between the front and back end, enabling data retrieval and manipulation.

    - Ensure secure and efficient handling of API requests and optimise performance.

- **Authentication and Authorization:**
    - Implement user authentication (e.g., via JWT, OAuth, or session-based login) to manage user sessions and permissions.
    - Manage different data access levels based on user roles (admin, guest, etc.).

- **Performance Optimization:**

    - Write efficient, optimised code to handle high volumes of data and requests.
    - Caching techniques (e.g., Redis, Memcached) can reduce load times and improve performance.
    - Load balancing, database indexing, and query optimisation.
- **Security:**

    - Implement security best practices, such as **data encryption**, **password hashing**, and input validation, to prevent SQL injection, XSS attacks, and other vulnerabilities.
    - Manage SSL certificates and apply HTTPS for secure communication.

Key Technologies & Tools:

- **Programming Languages:**

    - **JavaScript (Node.js):** Popular for both front-end and back-end development.
    - **Python (Django, Flask):** Django is known for rapid development, while Flask is a more lightweight framework.
    - **Ruby (Ruby on Rails):** Known for convention over configuration, great for rapid prototyping.
    - **PHP (Laravel, Symfony):** Used for dynamic websites and web applications.
    - **Java (Spring Boot):** Often used in large-scale enterprise applications.
    - **Go, C# and other languages** are also used in backend systems, especially for performance-critical applications.
    - **Spring Boot (Java):** A framework for building production-ready web applications in Java.

App Developer - Fabrizio Morales

Key Responsibilities

- **App Design & Development**: Creating, coding, and deploying mobile applications from scratch or improving existing applications.

- **User Interface (UI) and User Experience (UX) Design**: Collaborating with designers to implement intuitive and visually appealing interfaces that enhance user engagement.

- **Performance Optimization**: Ensuring apps are optimised for efficiency and speed, especially for mobile platforms with limited resources.

- **Debugging & Troubleshooting**: Identifying and fixing bugs in the app to improve functionality and user experience.

- **Testing**: Conduct thorough testing, including unit tests, integration tests, and user testing, to ensure the app is stable and reliable.

- **Maintenance and Updates**: Regularly update the app to add new features, fix issues, and adapt to the latest operating system versions or hardware updates.

- **Collaboration**: Working with a team that may include designers, product managers, backend developers, and QA testers to deliver a cohesive product.

- **Documentation**: Writing technical documentation to help other developers understand and maintain the codebase.

- **Key Skills and Technologies**
- **Programming Languages**:
  - **For iOS**: Swift and Objective-C.
  - **For Android**: Kotlin and Java.
  - **Cross-platform**: Dart (for Flutter), JavaScript/TypeScript (for React Native), or C# (for Xamarin).

- **Frameworks and Libraries**:
  - **iOS**: UIKit, SwiftUI.
  - **Android**: Jetpack, Android SDK.
  - **Cross-platform**: Flutter, React Native, Xamarin.

- **Backend Skills** (optional but beneficial): Knowledge of backend development (Node.js, Firebase, etc.) or APIs to integrate app functionalities with servers.
- **Version Control**: Familiarity with Git and GitHub/GitLab/Bitbucket for collaborative work.

- **Database Management**: Using databases like SQLite, Core Data (iOS), Realm, or Firebase for app data storage.
- **APIs and Networking**: Experience with RESTful APIs, GraphQL, and handling network requests.

IT System Engineer – Paul Hayward

Key Responsibilities

- System Design & Implementation: Planning and deploying IT infrastructure, including servers, networks, and storage solutions.
- System Maintenance & Troubleshooting: Monitoring systems, diagnosing issues, and resolving problems to maintain system availability and performance.
- Network Management: Configuring, maintaining, and troubleshooting network equipment (routers, switches, firewalls) and ensuring network security and connectivity.
- Security & Compliance: Implementing security protocols and ensuring systems comply with industry standards and regulations, including data protection, antivirus, and patch management.
- Backup & Recovery: Setting up and managing backup solutions to ensure data can be restored during a failure, disaster, or security breach.
- Automation & Scripting: Using scripting languages to automate repetitive tasks, deployments, or configurations to improve efficiency.
- Documentation: Creating and maintaining documentation for configurations, procedures, and troubleshooting steps to ensure knowledge sharing and continuity.
- Collaboration & Support: Working with other IT staff, departments, and end-users to provide technical support and solve IT-related issues.
- Key Skills and Technologies

- Operating Systems:
    - Servers: Experience with Windows Server, Linux (RedHat, Ubuntu, CentOS), and sometimes macOS for specific needs.
    - Desktops: Proficiency with Windows, macOS, and Linux desktop environments.
- Networking:

    - Knowledge of TCP/IP, DNS, DHCP, VPNs, and network security protocols.
    - Experience with networking hardware and software, such as Cisco, Juniper, and Ubiquiti devices.

- Virtualization & Cloud Computing:

    - Familiarity with VMware, Hyper-V, or KVM for virtualization.
    - Knowledge of cloud platforms like AWS, Azure, or Google Cloud.

- Storage Solutions:
    - Configuring and managing storage systems, such as SAN (Storage Area Network), NAS (Network Attached Storage), and cloud storage.
    - Understanding RAID configurations and backup solutions.

- Security:
    - Implementing firewalls, antivirus solutions, intrusion detection systems (IDS), and encryption.
    - Familiarity with security standards (like ISO 27001, GDPR) and tools to ensure compliance.

- Automation & Scripting:
    - Proficiency in scripting languages, such as PowerShell, Bash, or Python, to automate tasks and streamline operations