PLACEMENTDOST

Full Stack Web Development Assessment questions

Note:

Please be advised that these assessment inquiries are designed to accommodate interns with diverse skill levels, ranging from novices to seasoned analysts. Should you encounter any challenging questions, you are encouraged to seek solutions independently or reach out to us for assistance at intern@Placementdost.com

Best wishes for success in completing the assessment!

1. **Set Up Development Environment:**
   - Assist in setting up a development environment by installing necessary tools and dependencies, including text editors, version control systems (e.g., Git), and local web servers (e.g., XAMPP, WAMP, or MAMP).

2. **Frontend Development Tasks:**
   - Create responsive layouts using HTML and CSS frameworks like Bootstrap or Tailwind CSS.
   - Implement frontend interactivity with JavaScript and frontend frameworks/libraries like React, Angular, or Vue.js.
   - Develop user interface components such as forms, buttons, and navigation menus.

3. **Backend Development Tasks:**
   - Design and implement RESTful APIs using backend frameworks like Express.js (Node.js), Django (Python), or Laravel (PHP).
   - Set up and configure databases using SQL (e.g., MySQL, PostgreSQL) or NoSQL (e.g., MongoDB) databases.
   - Implement server-side logic for handling requests, processing data, and interacting with databases.

4. **Database Management Tasks:**
   - Design database schemas and define data models based on application requirements.
   - Write SQL queries to retrieve, insert, update, and delete data from databases.
   - Optimize database performance by creating indexes, optimizing queries, and managing database transactions.

5. **Authentication and Authorization:**
    - implement user authentication and authorization functionalities using technologies like JWT (JSON Web Tokens) or OAuth.
    - Set up user registration, login, and password management features.
    - Define user roles and permissions to control access to different parts of the application.

6. **API Integration:**
    - Integrate third-party APIs (e.g., payment gateways, social media APIs) into the application.
    - Handle API requests and responses using client-side JavaScript or backend server logic.
    - Securely manage API keys and authentication tokens to ensure data privacy and security.

7. **Performance Optimization:**
    - Analyze and optimize frontend performance by minimizing page load times and reducing render-blocking resources.
    - Implement caching mechanisms (e.g., browser caching, server-side caching) to improve application speed and responsiveness.

Projects:

E-commerce Website: Build an e-commerce website with features for browsing products, adding items to cart, and completing purchases. Implement user authentication, product catalogue management, shopping cart functionality, payment processing, and order management.

Task Management Application: Develop a web-based task management application where users can create, assign, and track tasks. Implement features such as user authentication, task CRUD operations, task assignment, status updates, and notifications.