



Technical Test: Node.js Developer with TypeScript, Fastify, Prisma, and Monorepo

Task 1: Advanced Monorepo Setup (25 points) [Optional]

- Set up a monorepo structure using a tool like Lerna or Yarn Workspaces.
- Consider creating separate packages for different concerns (e.g., API, services, utilities).

Task 2: Advanced Server Setup (20 points)

- Implement a Fastify application using TypeScript for enhanced type safety.
- Organize the codebase to support scalability and maintainability.

Task 3: Advanced Database Integration with Prisma and PostgreSQL (30 points)

User Management (15 points)

- Integrate Prisma for database access with a PostgreSQL database and define a User model.
- Implement CRUD operations for user management (create, read, update, delete).
- Include input validation and error handling for user-related operations.

Relationships and Data Validation (15 points)

- Extend the data model to include relationships between entities (e.g., users and posts).
- Implement advanced data validation using Prisma's constraints.

Task 4: Authentication and Authorization (20 points)

- Implement user authentication using JWT (JSON Web Tokens).
- Create middleware for route authorization based on user roles and permissions.
- Secure sensitive operations (e.g., user profile updates) behind authentication.

Task 5: Advanced API Features (20 points)

File Upload (10 points)

- Implement an endpoint for users to upload profile pictures.
- Store uploaded files securely and efficiently (consider using cloud storage if applicable).

Real-time Notifications (10 points)

- Implement a real-time notification system using WebSocket or a similar technology.
- Notify users of relevant events (e.g., new messages, updates).

Task 6: Testing and Documentation (15 points)

- Write unit tests for critical functions and endpoints using a testing library (e.g., Jest).
- Provide comprehensive API documentation using tools like Swagger or Fastify Swagger.

Task 7: Monorepo and Separate Packages (25 points)

[Optional]

- If you've chosen to set up a monorepo, ensure that the structure is well-organized.
- Create separate packages for different concerns (e.g., API, services, utilities).
- Demonstrate how changes in one package affect others.

Submission Guidelines:

Share the link to the Git repository containing your code.

Provide a detailed README with instructions, architecture overview, and any additional information.

Ensure that your API is deployed, and share the deployment link (e.g., Heroku, AWS).

