DevifyX

Node.js Job Assignment

Learning Progress Tracker

Assignment Description: User's course progress
Assignment Deadline: 7 Days
Assignment Submission Form: `https://forms.gle/LAvLWFmHRLXswwsx5`

1 Objective

Develop a scalable and maintainable Learning Progress Tracker web application using Node.js. The system should allow users to enroll in courses, track their progress, and visualize their learning journey. The goal is to demonstrate your ability to design, implement, and document a backend service with RESTful APIs, data persistence, and user authentication.

2 Core Features

1. **User Registration and Authentication:** Implement secure user registration, login, and logout functionalities.
2. **Course Management:** Allow administrators to create, update, and delete courses.
3. **Course Enrollment:** Enable users to browse available courses and enroll in them.
4. **Progress Tracking:** Track and update users' progress for each enrolled course (e.g., completed modules/chapters).
5. **Progress Visualization:** Provide endpoints to retrieve and display users' progress (e.g., percentage completed).
6. **RESTful API Design:** Design clean and well-documented RESTful endpoints for all functionalities.
7. **Data Persistence:** Store all data using a database (e.g., MongoDB, PostgreSQL).
8. **Role-based Access Control:** Differentiate between regular users and administrators with appropriate permissions.

3 Bonus Features

- Implement JWT-based authentication and refresh tokens.
- Add support for course completion certificates (PDF generation).
- Integrate email notifications for enrollment and course completion.

- Implement API rate limiting and request validation.
- Add support for user profile management (e.g., avatar upload, bio).

4 Technical Requirements

- Use **Node.js** (v14 or above) with **Express.js** or a similar framework.
- Database: MongoDB (preferred) or PostgreSQL.
- Use environment variables for configuration (e.g., `dotenv`).
- Proper error handling and input validation.
- Code must be structured and modular (use MVC or similar architecture).
- API documentation using Swagger/OpenAPI or similar.
- Use Git for version control. Include a `README.md` with setup instructions.
- Write at least basic unit tests for critical endpoints (e.g., Jest, Mocha).

5 Deliverables

- Source code repository (GitHub, GitLab, etc.).
- `README.md` file with setup instructions, API documentation, and usage examples.
- Database schema/model definitions.
- Sample data/scripts for seeding the database.
- Postman collection or equivalent for API testing.
- (Optional) Deployed demo link or instructions for local deployment.

6 Submission

- Submit your assignment using the following form: `https://forms.gle/LAvLWFmHRLXswwsx5`
- Ensure your repository is accessible (public or with appropriate access granted).
- Include your name and contact information in the submission form.

7 Evaluation Criteria

- **Functionality:** Completeness and correctness of implemented features.
- **Code Quality:** Cleanliness, modularity, and adherence to best practices.
- **API Design:** RESTful principles, documentation, and usability.
- **Security:** Proper authentication, authorization, and input validation.
- **Testing:** Presence and coverage of unit tests.
- **Documentation:** Clarity and completeness of the `README.md` and API docs.
- **Bonus:** Implementation of any bonus features.
- **Timeliness:** Submission within the deadline.

Good luck! We look forward to your submission.
— **DevifyX Team**
Click here to read our Terms and Conditions