HYRUP

Backend Developer - Technical Assignment

Assignment Overview

Welcome to HYRUP's technical assessment! This assignment is designed to evaluate your full-stack development skills, with a focus on backend architecture, security best practices, and modern authentication patterns.

Duration: 48-72 hours from receipt

Objective

Build a Student Management System with secure authentication and authorization. You will demonstrate your ability to design data models, implement secure authentication flows, and create RESTful APIs.

Technical Requirements

1. Student Model Design

Create a comprehensive Student model with fields that align with HYRUP's vision. The choice of fields is left to your creativity, but should include:

- Essential identification fields (name, email, student ID, etc.)
- Academic information (course, year, enrollment date, GPA, etc.)
- Contact details (phone, address, emergency contact)
- Additional fields that showcase your understanding of student management systems

Note: *The model design should reflect thoughtful consideration of real-world student management needs and demonstrate proper data modeling principles.*

2. Authentication System

Registration (Sign Up)

- Accept email and password from the user
- Implement proper email validation
- **CRITICAL:** Password must be encrypted/hashed before storing in the database (use bcrypt, argon2, or similar)
- Return appropriate success/error responses
- Generate and return JWT token upon successful registration

Login (Sign In)

- Accept email and password
- Verify credentials against stored (encrypted) password
- Generate and return JWT token upon successful authentication
- Implement proper error handling for invalid credentials

JWT Implementation

- Use JWT (JSON Web Tokens) for session management
- Include appropriate claims in the token payload (user ID, email, role if applicable)
- Set reasonable token expiration time
- Implement middleware to protect routes that require authentication

Technology Stack Recommendations

You are free to use any modern technology stack, but here are some recommendations:

Backend

- Node.js (Express) or Python (Django, Flask, FastAPI)
- Java (Spring Boot)

Database

- PostgreSQL, MySQL, MongoDB
- Use an ORM/ODM (Prisma, TypeORM, Sequelize, Mongoose, SQLAlchemy)

Security

- bcrypt, bcryptjs, or argon2 for password hashing
- jsonwebtoken or similar library for JWT

Evaluation Criteria

Criterion	What We're Looking For
Security Implementation	Proper password encryption, secure JWT implementation, protected routes
Data Model Design	Thoughtful field selection, proper data types, relationships, validation
Code Quality	Clean, readable, well-organized code with proper structure

Error Handling	Comprehensive error handling with meaningful messages
API Design	RESTful principles, proper HTTP methods and status codes
Documentation	Clear README with setup instructions and API documentation

Deliverables

1. **Complete source code** hosted on GitHub (public or private repository with access granted)
2. **README.md** including:
- Project setup instructions
- Environment variables required
- Database setup/migration instructions
- API endpoint documentation
- Student model field explanations
- Any assumptions or design decisions made

3. **Postman Collection or API documentation** (optional but highly recommended)
4. **Database schema/ER diagram** (optional but shows thoroughness)

Bonus Points

Stand out from other candidates by implementing:

- Input validation and sanitization
- Refresh token implementation
- Rate limiting on authentication endpoints
- Unit tests for critical functions
- API versioning (e.g., /api/v1/students)
- Pagination for student list endpoint
- Search and filter functionality
- Docker containerization
- Environment-specific configurations (development, production)
- Logging and monitoring setup

Submission Guidelines

5. Submit your GitHub repository link via email (**hyrup.career@gmail.com**)
6. Ensure the repository is accessible (public or invite provided)
7. Include a .env.example file showing required environment variables

8. Submit within the specified timeframe

9. Use of AI is permitted, but it should not be relied upon entirely. You must be able to understand, explain, and present the concepts independently.

Important Notes

- **Security is paramount:** Never store plain text passwords. All passwords must be hashed.
- **JWT must be properly implemented:** Use secure secrets, appropriate expiration times, and validate tokens correctly.
- **Student model creativity counts:** Your field choices should demonstrate understanding of real-world applications.
- **Code quality matters:** Write code as if you're collaborating with a team. Clear, maintainable, and well-documented.
- **Questions are welcome:** If you need clarification, don't hesitate to reach out. We value communication.

Questions?

If you have any questions about the assignment, please reach out to the HYRUP recruitment team. We're here to help!

Good luck! We look forward to reviewing your submission.

— The HYRUP Team