

Technical Assignment

Backend Developer

Objective

This test evaluates your skills in **Node.js**, **Express.js**, and **MongoDB** by assessing your ability to build a RESTful API, handle authentication, and manage database operations.



Instructions

- Use **Node.js with Express.js** for the backend.
- Use **MongoDB** as the database (can use **MongoDB Atlas** or **local instance**).
- The test should be completed in **3-4 hours**.
- Code should be structured, commented, and follow best practices.
- Submit your code via a GitHub repository or ZIP file.



Task: Digital Voucher Management API

Scenario

You are hired to develop a **digital voucher management system** for an online platform. Users can **purchase, redeem, and manage vouchers** using this system.



Features Breakdown:

1 User Authentication & Authorization

- Implement **JWT-based authentication**.
- Users should be able to **register, login**, and **access protected routes**.
- Two user roles:
 - **Admin**: Can create, update, and delete vouchers.
 - **User**: Can purchase and redeem vouchers.

2 Voucher Management

- Implement CRUD operations for vouchers:
 - **Create** a new voucher (Admin only).
 - **Retrieve** available vouchers (Public).
 - **Update** voucher details (Admin only).
 - **Delete** a voucher (Admin only).
- Each voucher should have:
 - name (string, required)
 - code (unique string, auto-generated)

- value (number, required)
- status (enum: ["active", "redeemed", "expired"], default: "active")
- expiryDate (date, required)
- createdAt (auto-generated)

3 Purchasing & Redeeming Vouchers

- Users can **purchase vouchers** (reduce their balance).
- Users can **redeem vouchers** (mark as redeemed if not expired).
- Maintain a **transaction history** for purchases and redemptions.

4 Database Schema (MongoDB)

- **User Model:** (name, email, password, role, balance)
- **Voucher Model:** (name, code, value, status, expiryDate)
- **Transaction Model:** (userId, voucherId, type: ["purchase", "redeem"], date)

Bonus Features (Optional, Extra Points)

- ♦ Implement a **voucher expiry system** (auto-expire outdated vouchers).
- ♦ Add **filters & pagination** to retrieve vouchers.
- ♦ Implement **unit tests** using Jest or Mocha.

Submission Guidelines

- A **GitHub repository** or **ZIP file** with the source code.
- A **README file** explaining how to install and run the project.
- Postman **API documentation** or **Swagger UI** for API endpoints.
- Send all files to [careers@t7dgaming.com]

Evaluation Criteria

- ✓ **Code Structure & Readability**
- ✓ **Correctness of API Endpoints**
- ✓ **Database Design & Queries**
- ✓ **Security (JWT, Password Hashing)**
- ✓ **Error Handling & Validations**
- ✓ **Bonus Features (if implemented)**

Sample API Endpoints

Method	Endpoint	Description	Access
POST	/api/auth/register	Register a new user	Public
POST	/api/auth/login	User login & token issue	Public
GET	/api/vouchers	Fetch all vouchers	Public
POST	/api/vouchers	Create a voucher	Admin
PUT	/api/vouchers/:id	Update a voucher	Admin
DELETE	/api/vouchers/:id	Delete a voucher	Admin
POST	/api/vouchers/purchase	Purchase a voucher	User
POST	/api/vouchers/redeem	Redeem a voucher	User

Deadline

 You have **24 hours** to complete and submit the task.