Emerging Technology Lab

Assignment 7

Development and Testing of REST API with NodeJS, ExpresJS, and MongoDB

Create a NodeJS application to provide endpoints for a project called **Campus Trade.** The objective of the project is to trade used items (like book, calculator, Engineering Equipment, Sports equipment etc) with in the campus. There are two collections **Users** and **Listings**. Complete the below tasks to provide endpoints for the project.

Install required packages like – express, mongoose, nodemon and perform the following operations. Handle asynchronous behaviour and exceptions properly.

1. Define schema and model for **User,** with field name, mobile, email, sic number and password. All fields must be required and sic, mobile, and email must be unique.
2. Create a file to establish the connection with MongoDB and select the database.
3. Create a controller file for **User** and add functionalities for –
    a. Add a user.
    b. Retrieve all users.
    c. Retrieve a single user based on ID
    d. Retrieve a single user based on SIC/email/mobile
    e. Update a student based on SIC
    f. Delete a student based on SIC
4. Create a router file to connect with the User controller methods. Link the router to entry point of the project and test all the endpoints using Postman.

Create an .env file to store all the project's environment variables and modify the code accordingly. Install **multer** and configure it properly to upload files to server, before proceed to the next phase.

5. Define schema and model for **Listings,** with fileds - itemName, category, description, condition, price, imageName, status, owner's id. All fields are required.
    a. Category should be one of - Book, Engineering Equipment, Stationery, Electronics, Sports Equipment, Clothing, and Other.
    b. Condition should be one of – As New, Good, Poor.
    c. Status should be either Available or Sold.
    d. Owner's Id is a reference of Users collection.
6. Create a controller file for Listings and add functionalities for –
    a. Add a listing.
    b. Retrieve all listings.
    c. Retrieve a single listing details (including owner details) using id.
    d. Update listing details based on id.(without image)
    e. Update listing image based on listing id.
    f. Delete a listing details based on id.
7. Create a router file to connect with the Listing controller methods. Link the router to entry point of the project and test all the endpoints using Postman.