Assignment Guideline: Backend Node.js Module - URL Shortener and Redirection

Project Overview:

- Design and implement a backend application using Node.js that serves as a URL shortener and redirection service. The application should allow users to register, login, create, edit, and delete short URLs. Additionally, it should track URL visit counts and provide analytics for the URL owners.

User Roles:

- APP_USER: Regular authenticated users.
- BONUS: ADMIN: Admin users with override privileges.

REQUIRED USER STORIES:

1. User Registration and Authentication (10 marks)
    a. Users can register with their email and password.
    b. Email validation is required before login.
    c. Non-validated accounts cannot login.
    d. Authenticated users receive an authentication token upon successful login.
2. URL Management (15 marks)
    a. Authenticated users can view all URLs they've created.
    b. Users can create a short URL that redirects to a destination URL.
    c. Short URLs generated must be unique and randomised.
    d. Authenticated users can edit and delete their own URLs.
3. Public Access and Redirection (10 marks)
    a. Short URLs are accessible to the public and trigger redirection to the destination URL.
    b. Visit counts for each short URL are stored in the database.
4. Analytics (10 marks)
    a. Owners can view analytics for their URLs, including visit counts.

BONUS USER STORIES:

1. Admin Role (5 marks)
    a. Implement an admin role with override privileges for all actions.
2. URL Activation Status and 404 Page (5 marks)
    a. Add functionality to classify short URLs as active or inactive.
    b. Active URLs redirect to the destination, while inactive URLs return a predefined HTML page indicating 404.
3. User can generate report for all of the list and generate a CSV

Marking Scheme:

- Total marks: 55

Evaluation Criteria:

1. Database Design and ERD:
    a. Design of the database schema reflects the requirements outlined in the user stories. (15 marks)
    b. Entity-Relationship Diagram (ERD) accurately represents the relationships between database entities. (5 marks)
    c. Proper normalisation and optimization of the database schema for efficient data storage and retrieval. (5 marks)
2. Implementation Quality:
    a. Code readability and structure (10 marks)
    b. Proper use of Node.js best practices (10 marks
3. Functionality Completeness:
    a. All required user stories implemented (20 marks)
    b. Bonus user stories implemented (10 marks)
4. Security and Validation:
    a. Proper validation of user inputs (5 marks)
    b. Secure authentication and authorization mechanisms (10 marks)
5. Public app performance:
    a. Efficient public retrieval of URL data (5 marks)
    b. Proper handling of URL visit counts (5 marks)
6. Bonus
    a. Admin functionality (10 marks)
    b. URL Activation Status (10 marks)
    c. CSV report generation (10 marks)

Submission Instructions:

Submit your Github repository project source code along with documentation outlining the project structure, API endpoints, and any setup instructions.

Ensure all code is well-commented and includes appropriate documentation for setting up and running the application.

Important Note:

Plagiarism will not be tolerated and will result in immediate disqualification.
If you have any questions or need clarification on the requirements, feel free to reach out for assistance.