University of Central Punjab

Faculty of Information Technology

Web Application Development

Assignment 01

Course Code: SESE3143	**Instructor:**
Marks: 20	**Semester:** Fall 2025

Deadline of Assignment: [26 Oct, 2025, 11:59PM]

Instructions:

- **Submission:**
  - This Assignment is individual.
  - All submissions will be on UCP Portal only. Upload zip file with complete documentation
  - Ensure the code is functional and well-documented with comments
  - You have to submit soft copy.
    - **For soft copy submission**, **upload a zip file containing all the required code files + MS-Word document containing the code and screen shot of the output** to the UCP Portal before deadline
- **Plagiarism Policy and Late Policy:**
  - Strictly no plagiarism; ensure all work is original. In case of plagiarism, zero marks will be awarded.
  - In case of late submission, 25% marks will be deducted.
- **Evaluation Criteria:**
  - Assignment 01 will be evaluated on the basis of Quiz 01. (50%-50%)

Submission by:

Name	Roll Number

Section:______

CLO 1	Cognitive Level	Keywords
Students will be able to apply foundational and advanced concepts of Node.js, Express.js, and MongoDB to build robust backend solutions for web applications.	C3	Apply

Assignment: Library Management System API Development

Your task is to **apply** node.js and express.js to create a simple RESTful API that manages information about books in a library. This assignment will help you practice your skills in building APIs, working with static data, and understanding how to structure your applications.

1. Assignment Details:

Scenario: Library Management System

You are developing an API for a library that needs to provide information about its books to clients (like a front-end application or other services). Your API will allow users to retrieve data about various books in the library.

2. Requirements:

3.1 Project Setup:

1. Initialize a new Node.js project using npm init to create your package.json file. Make sure to add your project information in the package.json file.
2. Install the Express.js framework to create your API.
3. Install Nodemon to help with automatic server restarts during development.

3.2 Static Data Structure:

1. Create a separate file named data.js to define an array of objects representing books in the library. Each book object should include the following properties:

   - **id:** A unique identifier for the book (number).
   - **title:** The title of the book (string).
   - **author:** The author of the book (string).
   - **is_available:** A boolean indicating whether the book is available for checkout.

2. Use module.exports in data.js to export the books array.
3. Import this array into your index.js file and access the data from there.

3. API Endpoints:

Implement the following GET endpoints in your API:

1. **List of All Books**: Retrieve a complete list of all books in the library (e.g., `GET /api/books`).
2. **List of Available Books**: Retrieve a list of all books that are currently available for checkout (e.g., `GET /api/books/available`).
3. **List of Issued Books**: Retrieve a list of books that have been issued to members (e.g., `GET /api/books/issued`).

4. **Find Books by Author**: Retrieve a list of all books by a specific author (e.g., `GET /api/books?author=AuthorName`).

5. **JSON Responses:**

   Ensure that your API returns responses in JSON format for all endpoints, making it easy for clients to consume the data.

6. **Using Nodemon:**

   Set up Nodemon in your project to enable automatic server restarts when you make changes to your code. This will improve your development experience.

7. **Documentation:**

   Create a `README.md` file that explains how to set up and run your project. Include:

   - Instructions for installing dependencies.
   - How to start the server.
   - A list of the API endpoints with examples of requests and responses.

8. **Submission Guidelines:**

Submit your assignment on the ODOO portal as a zip file containing the entire project directory, including:

- `index.js`: Your main server file.
- `data.js`: The file containing your static data structure.
- `package.json`: Your project configuration file.
- `README.md`: Documentation for your API.

Ensure that your code is well-organized, properly commented, and adheres to best practices.

9. **Due Date:**

Your submission must be uploaded by **October 26, 2025, at 11:59 PM**. Late submissions may not be accepted.

10. **Evaluation Criteria:**

- **Functionality**: All endpoints must work correctly and return the expected data.
- **Code Quality**: Code should be clean, well-structured, and properly commented to enhance readability.
- **Documentation**: The README file should provide clear instructions on how to run the project and use the API.