SMART TRAINERS

LESSON PLAN FOR BACKEND WEB DEVELOPMENT COURSE

COURSE TITLE: **Backend Web Development**	COURSECODE: **BWD 001**	CREDIT HOURS: **01**	DURATION: **08 Weeks (03 Classes Each Week)**
COURSE INSTRUCTOR: **Sakhawat Ali Larik**			
Batch:	Course Starting Date:	Course Suspension Date:	

COURSE LEARNING OUTCOMES: Upon successful completion of the course, the student will be able to:

- **Design and Develop** robust server-side applications using NodeJS and ExpressJS.
- **Implement Secure Systems** including authentication, authorization, and data encryption.
- **Manage Complex Databases** using MongoDB, including CRUD operations and query optimization.
- **API Development and Maintenance** for seamless integration and management of data for frontend.
- **Optimize Performance** through caching, load balancing, and efficient server logic management.

Note: This course requires prior knowledge about Web Development.

CLO	DESCRIPTION	Learnings
1. Server Architecture	Understand how servers are created, how server routing, and request handling work.	1. NodeJS Basics 2. Express JS 3. Server Request and Responses
2. Security & Auth	Implement industry-standard security and user identity verification.	4. JWT, OAuth, & Encryption 5. RBAC and POLA
3. Database Logic	Understanding the data logics, manage data storage, retrieval, and validation effectively.	6. Relational Vs Non-Relations Databases 7. Database Management Systems 8. MongoDB & Mongoose ODM or Mongo Client
4. API Development & Testing	Manage data storage, retrieval, and validation for integration with others	9. RESTful API Development and maintenance with Postman 10. Implement Server best logics and rules for security

LESSON CONTENTS AND ASSOCIATED CLO(s)				
Contents	**CLO / Week**	**Marks %**	**Delivery Methods**	**Assessment Methods**
1. Introduction of web development • How internet and websites work • The client-server architecture • Types of web development • The website evaluation • Basics of web browsers & servers **2. JavaScript Language Basic recap** • Introduction of JS • The TypeScript Concepts • The Node JS Concepts **3. Node JS in detail** • Introduction to Node JS Programming • Node JS File Handling • Node JS Creating Server Environment	01	5%	Online Lectures & Practical work	Practical Project
4. Request, Response and Rendering • Routing and Rendering Basics • The Request Methods • The Response Types • The Request Header in Details • The Response Header in Details **5. Express JS Framework** • Introduction to Express JS • Mapping requests to specific routes and rendering • *Request parsing*: Parsing HTTP requests, including headers, query parameters, and bodies • *Request validation*: Validating incoming requests, including data validation and authentication. **6. Server-Side Vs Client-Side and Template Engines** • The difference between server-side programming and client-side programming specially in JS • The EJS template engine for client-side programming in Express JS	02	5%	Online Lectures & Practical work	Practical Project

7. Databases for Data Storage and Retrieval • Introduction to Databases • Database Management Systems • The SQL Vs NoSQL **8. MongoDB (NoSQL Database)** • Introduction to MongoDB • MongoDB CLI and Compass Setup • MongoDB Local and Atlas Setup • MongoDB CRUD Operations • MongoDB Database Queries **9. MongoDB with Express JS** • Introduction to Mongoose • Introduction to Mongo Client • Data Validation and Schemas	03	10%	Online Lectures & Practical work	Practical Project

10. Mongoose CRUD Operations with Express JS • Schema Design with Mongoose • CRUD operations with Mongoose on MongoDB Databases **11. RBAC and POLA with DB Structure** • Creating RBAC based Databases and Tables • Login and Registration Process • Controlling web apps privacy with RBAC and POLA rules • Verifying users with credentials **12. Session and Cookies Management** • Introduction to Session Storage • Introduction to Cookie Storage • Sharing data with sessions and cookies	04	10%	Online Lectures & Practical work	Practical Project
13. Server and Data, Security and Privacy • *Input validation*: Validating user input to prevent attacks. • *Data validation*: Validating data before storing or retrieving it. • *Authentication*: Verifying users through tokens, or OAuth. • *Authorization*: Controlling access to resources based on user roles, permissions, or other criteria • *Encryption*: Encrypting data and sensitive information. • *Rate limiting*: Limiting the number of requests from a single IP or user to prevent abuse. • *Error handling*: Catching and handling errors, including logging and returning error responses.	05	20%	Online Lectures & Practical work	Practical Project

14. File Handling and Server File Uploads/Downloads • Uploading files to server and Database • Introduction to Multer • Multer Single File Upload • Multer Multiple File Upload • Restrict File Uploads (By Size, Error and Types) **15. The Concept of Middleware** • Introduction to common middleware • Creating custom middleware **16. The MVC Concept in Backend Development** • Introduction to MVC • Creating Models • Creating Views • Handling Routes • Creating Controllers	06	20%	Online Lectures & Practical work	Practical Project
17. API Development • Designing APIs with Express JS • The Cross Origin Resource Sharing • Restrict API usage and Call • Testing and Using APIs with POST • Understanding API tokens and API limits **18. The Complete RESTful API Development and Implementation with React JS frontend** **19. The Other Types of APIs** • RESTful API • SOAP API • JSON RPC • Web Socket • Web Hook	07	10%	Online Lectures & Practical work	Practical Project
20. The Complete Backend Projects • Backend Development with Express JS • API Development with Express JS • Database Implementation • File Handling • Server-side logics • Frontend template engine with EJS • Using all learnt concepts to create a complete React + Express JS project • Implementing MERN Stack approach **21. Course Closing** • Course Closing and Next Steps (Discussion)	08	20%	Online Lectures & Practical work	Practical Project