National University of Computer and Emerging Sciences, Lahore Campus

Course:	**Web Programming**	**Course Code:**	
Program:	**BS**	**Semester:**	**Fall 2025**
Out Date:	**01-Dec-25**	**Total Marks:**	**100**
Due Date:	**8-Dec-25**	**Weightage:**	**3**
Section:	**A/B**	**Page(s):**	**02**
Assignment:	**04**		

Instruction/Notes:	You are required to submit your assignment on Google Classroom. No late submissions will be accepted. For any query, please contact your TA. Penalties may include grade reductions or other consequences as determined by the evaluation guidelines. Plagiarism or AI-generated code is strictly prohibited. You must be able to fully understand and justify your implementation. **Be prepared to justify individual contributions.** **Submit a zipped folder containing files along with the link of your public GitHub Repository.** Any violation of the submission instructions, unequal contribution, or evidence of plagiarism will result in penalties.

Assignment Instructions

Extended Assignment: Build a Fully Functional MERN Application

Your Assignment 2 front-end (FE) must now be extended into a **complete, production-ready MERN stack application**.
This means:

- **M**ongoDB → database
- **E**xpress.js → backend server & API
- **R**eact.js → front-end UI
- **N**ode.js → runtime environment

Functional Requirements for the MERN Application

1. User Authentication & Authorization

- User registration & login via JWT
- Password hashing using **bcrypt**
- Access tokens + optional refresh tokens
- Role-based access control (e.g., user, admin)

- Route protection middleware
- Rate limiting & brute-force prevention

2. **CRUD Operations for Main Business Entity**

(Entity depends on your Assignment 2 project: posts, products, tasks, etc.)

- Create, Read, Update, Delete
- Server-side and client-side validation
- Pagination (?page= and ?limit=)
- Search and filtering (?search=, ?sort=)
- Ownership checks (only item creator can edit/delete)

3. **Dashboard & Profile Management**

- JWT-protected dashboard
- Data analytics (counts, comparison, maybe charts)
- User profile update
- Password reset/change flow

4. **API Requirements**

Your Node.js + Express backend MUST include:

- RESTful routes
- Input validation (Joi, validator.js, or custom)
- Response messages with correct status codes
- Error handling middleware
- Authentication middleware (JWT verify)
- Database model(s) using Mongoose
- Image upload
- Role-based access (admin/user)
- Email notifications

5. **Database Requirements**

- MongoDB collection for users and images
- Collection for your main entity
- Schema validation with Mongoose
- Proper relationships if needed (e.g., user owns posts)

6. **Deployment Requirements**

- Deploy FE on Vercel/Netlify

7. **Backend must include**

- MVC or service-layer architecture
- Modularized route/controller structure
- Error-handling middleware

- Async error wrapper (`catchAsync`)
- Environment variables via **dotenv**
- Mongoose schema & validation

8. **Security Requirements**
   - CORS configuration
   - Input sanitization (NoSQL injection prevention)
   - Rate Limiting

9. **Testing Requirements**

Add unit test for one POST and PATCH Api using at least **one** of the following:

- Unit tests (Jest)
- API testing (Supertest)
- E2E tests (Cypress)

Tips:

- Explore **Redis** and understand how modern systems use it for **caching, session management, rate limiting, and performance optimization**. If possible, try implementing basic caching for one of your API routes.
- Learn the basics of **Docker** and attempt to **containerize your MERN application**, including separate containers for the frontend, backend, and database. Understanding Docker early will help you with real-world deployment and DevOps tasks.
- Explore **ORMs and ODMs**, especially **Mongoose**, and SQL-based ORMs like **Sequelize**. Learn how they help structure and validate data models in backend systems and why they are essential in industry development.

You started this course with HTML. Now you're deploying containerized, database-driven MERN apps.
Congrats — you are now dangerous.