B8IT146 – Web and Cloud Application Development

100% Continuous Assessment Project (Student Brief)

Overview

This project counts for **100% of your module grade**. You will work in groups of **2–3 students** to design and develop a **full-stack web application** using:

- ReactJS (Front-End)
- NodeJS + Express (Back-End)
- MySQL (Database)
- Bootstrap (Optional for styling)

You may choose: *E-Commerce site, booking portal, learning portal, shop, service system, or any approved idea.* For a full list of approved project ideas, see Appendix A.

Each student must submit individually on Moodle. An **individual oral defence** worth 50% will take place in Week 13.

Key Deadlines

- **Week 10 – Front-End Draft Submission (Feedback Only)**
- **Week 13 – Final Submission + Individual Oral Defence**

1. Project Requirements

Your application must include:

- At least **8 React pages/views**, such as:
  - Home
  - Login
  - Registration
  - Dashboard / Welcome page
  - Catalogue (List of items from the database)
  - Item Details / Gallery
  - Cart / Checkout or equivalent action

- Fully functional **React front-end** with routing, components, props, and state.
- **NodeJS + Express back-end** with REST API endpoints.
- **MySQL database** with minimum 4 entities and proper relationships.
- **Authentication** (login, registration, tokens/sessions).
- **Client-side and server-side validation**.

2. Technology Stack

You **must use**:

- HTML5, CSS3
- ReactJS
- NodeJS + Express
- MySQL

Optional but allowed:

- Bootstrap or any CSS framework
- Axios or Fetch
- JWT authentication

Not allowed:

- PHP
- Server-side rendering templates (EJS, Handlebars)
- jQuery-only projects

3. Detailed Deliverables

Task 1 – Front-End Development (Week 10)

Worth: 15 marks (feedback only; final marks awarded Week 13).

- React project setup.
- Navigation and routing between pages.
- UI mock-ups using components and Bootstrap (optional).
- Client-side validation (required fields, email format, etc).
- Submit:
  - Zipped React project (front-end only).
  - Short description of project idea and front-end structure (1–2 pages).

Task 2 – Back-End Development (NodeJS + Express)

Worth: 20 marks

- NodeJS + Express server created as a separate project.
- REST endpoints such as:
  - POST /api/auth/register
  - POST /api/auth/login
  - GET /api/products
  - GET /api/products/:id
  - POST /api/cart or /api/orders
- Server-side validation (e.g., registration form).
- Authentication using sessions or JWT.
- Connect to MySQL and perform CRUD operations.
- Integrate API with React front-end.

Task 3 – Database (MySQL)

Worth: 10 marks

- Minimum 4 entities (e.g. User, Product, Order, OrderItem).
- Clear ERD showing: PKs, FKs, and relationships.
- Database implemented in MySQL.
- Sample data inserted.
- NodeJS back-end must read/write to the DB.

Task 4 – Documentation + Video Demo

Worth: 5 marks

- Final written report (3–6 pages):
  - Introduction and system overview
  - Screenshots of final UI
  - React front-end explanation
  - NodeJS back-end explanation
  - ERD + explanation
- Video demo (5–8 minutes) showing full system.

4. Individual Oral Defence (Week 13)

Worth: 50 marks

Each student must individually demonstrate:

- Understanding of React components, routing, state and props.
- Understanding of Express routes, controllers, and API structure.
- How the system connects React → NodeJS → MySQL.
- Your personal contribution to the project.
- Ability to answer follow-up technical questions.

5. Marking Scheme

The full marking scheme for the B8IT146 Web and Cloud Application Development Project is shown in Table 1. The assessment is divided evenly between group deliverables and an individual oral defence.

Table 1: B8IT146 Project Marking Scheme

Assessment Component	**Marks**
Group Work (50 Marks Total)	
Front-End Development (ReactJS UI, Navigation, Validation)	15
Back-End Development (NodeJS + Express API, Integration)	20
Database Design & Implementation (ERD + MySQL CRUD)	10
Documentation & Video Demonstration	5
Individual Work (50 Marks Total)	
Code Walkthrough (React, Express, Database Queries)	22
Technical Questions, Understanding, & Discussion	28
Total Marks Available	**100**

6. Suggested Project Timeline

- **Week 8:** Form groups, choose idea, get approval.
- **Week 9:** Build front-end pages with dummy data.

- **Week 10:** Submit front-end draft.
- **Week 10–12:** Complete ERD, MySQL, NodeJS API, integration.
- **Week 12–13:** Final testing, report, video, and oral preparation.

7. Getting Started (Checklist)

- Create GitHub repository for your team.
- Set up React project (Vite or CRA).
- Plan your pages and sketch simple wireframes.
- Set up NodeJS + Express server.
- Design ERD and create MySQL structure.
- Integrate React with API.
- Test all routes and pages thoroughly.

J. Minimum Requirements Checklist

All projects must include:

- React front-end with routing and components
- NodeJS + Express API
- MySQL database with minimum 4 entities
- CRUD functionality
- User authentication (login/register)
- Form validation (client-side & server-side)
- At least 8 pages/views

Appendix A — Approved Project Ideas

A. Project Selection Guide

To ensure your project is realistic, achievable, and aligned with the module's learning outcomes, you must select a project idea from the approved list below or propose your own idea for lecturer approval.

Your chosen system must be implemented using: **ReactJS (front-end), NodeJS + Express (back-end), and MySQL (database)**.

It must also include:

- at least 8 React pages/views,
- CRUD functionality,
- authentication (login & register),
- form validation (client & server),
- API integration between React and Express.

The following categories provide suggested project directions with examples and required core features.

B. Category 1 — E-Commerce & Online Shopping

E-Commerce systems allow users to browse, search, and purchase items.

Example ideas:

- Clothing store
- Electronics shop
- Plant nursery / gardening shop
- Car parts store
- Beauty & cosmetics shop
- Phone/laptop marketplace

Required features:

- Product list, filtering, and search
- Product details page
- Shopping cart
- Checkout simulation

- User authentication
- Optional: Admin page to add/edit products

C. Category 2 — Booking & Reservation Systems

These systems manage slots, time periods, appointments, or reservations.

Example ideas:

- Gym class booking
- Doctor/dentist appointment system
- Barber/hairdresser booking app
- Hotel or room reservation
- Sports pitch or court booking
- Cinema/theatre seat booking

Required features:

- Selection of dates/times/slots
- Booking confirmation
- Viewing past & active bookings
- Authentication
- Optional: Admin adding availability

D. Category 3 — Service-Based Platforms

These systems provide services rather than products.

Example ideas:

- Freelancer marketplace
- Car rental platform
- Home cleaning services
- Laundry pickup & delivery
- Pet grooming service
- Mini food delivery portal

Required features:

- Service catalogue
- Service details page
- Booking or ordering service
- Authentication
- Optional: Manage services panel

E. Category 4 — Learning & Educational Portals

Useful for lessons, content, or quizzes.

Example ideas:

- Online learning portal (courses/lessons)
- Study notes or flashcard system
- Coding tutorial platform
- Language learning mini-app
- Academic resources portal

Required features:

- Lessons list
- Lesson detail page
- User progress tracking (basic)
- Authentication
- Optional quizzes/assessments

F. Category 5 — Community, Social & Club Systems

Ideal for groups, clubs, and public sharing.

Example ideas:

- Sports club membership system
- Event discovery & joining platform
- Student society hub

- Recipe sharing portal
- Photo or art sharing gallery

Required features:

- Posts/events/items list
- Details page
- Save/favourite/join options
- Authentication
- Optional: User-generated content

G. Category 6 — Management & Admin Systems

Focus on internal control of data (simple, clean CRUD structure).

Example ideas:

- Library management system
- Warehouse/stock control
- Restaurant menu & staff management
- Patient records system (basic, non-medical)
- Office asset tracker

Required features:

- CRUD items/records
- Authentication
- Optional: Admin vs user roles

H. Category 7 — Creative & Media Portals

Great for design-oriented projects.

Example ideas:

- Photography showcase
- Music/album discovery
- Mini video library

- Portfolio builder
- Travel destination explorer

Required features:

- Catalogue of media/creative items
- Details page
- Favourites/collections
- Authentication

I. Category 8 — Student-Created Custom System

You may propose your own idea, provided it meets all criteria:

- Uses ReactJS + NodeJS/Express + MySQL
- Includes CRUD, authentication, and API integration
- Has at least 8 pages/views
- Is achievable within 5 weeks
- Is approved by the lecturer

This appendix provides a complete menu of project options. Students may select any idea as long as it meets the technical and functional requirements and is approved by the lecturer.

Appendix C — Assessment Rubric

Group Grading (50%)

Component	Absent (F)	Insufficient (1–59.49%)	Emerging (60–69.49%)	Satisfactory (70–76.49%)	Competent Mastering (77–100%)
Deliverable 1: Front-End Development (15 marks)					
Front-End Development	None of the following are present: HTML, Bootstrap and CSS elements and styling; form tags and validation; CSS liquid layout and box model; JS input and output; events and listeners; DOM elements; JavaScript functions; control statements; JS form validation; data type / parsing / conversion; JS library functions; jQuery event model and selectors; DOM modification with jQuery. Code does not run or display anything.	Only a few but not all of the above elements are implemented. Includes significant errors that cause the page not to display correctly.	Some but not all of the above elements are implemented. Code runs but may still have noticeable layout or functional issues.	Most but not all of the above elements are implemented. Code runs with no major errors and page layout generally displays correctly.	All of the abo elements a implemented (HTML/CSS/ validation, DOM, JS fun tions, cont statements, JS librari jQuery ever and DOM mo ification). Co runs with errors and pa layout displa correctly.
Deliverable 2: Back-End Development (20 marks)					

Continued on next pa

Component	Absent (F)	Insufficient (1–59.49%)	Emerging (60–69.49%)	Satisfactory (70–76.49%)	Competent Mastering (77–100%)
Back-End Development	None of the following are present: NodeJS and Express project with modules and dependencies; server-side interaction with HTML; NodeJS validation and required modules; HTTP web server using the built-in `http` module; correct use of request and response objects and callback functions; use of the URL property of the request object to check resources requested by the client; server-side listener for client requests. Code does not run or display anything.	Only a few but not all of the above elements are implemented. Code may run but includes significant errors and pages may not display correctly.	Some but not all of the above elements are implemented. Code runs with some errors and pages may display with noticeable issues.	Most but not all of the above elements are implemented. Server-side code runs and displays requested pages and results on the client side with minimal errors.	All of the abo elements a implemented (NodeJS/Exp setup, mo ules, valid tion, HT server, quest/respons handling, UI checking, liste ers). Code ru with no erro and request pages and sults displ correctly.
Deliverable 3: Database Design and Implementation (10 marks)					

Continued on next pa

Component	Absent (F)	Insufficient (1–59.49%)	Emerging (60–69.49%)	Satisfactory (70–76.49%)	Competent Mastering (77–100%)
Database Design & Implementation	No data layer implemented: no DAO, DB design, DB implementation, connection, or DB operations (adding, deleting, updating). Includes significant errors that cause the page not to display anything.	Contains minimal data layer: DAO, DB design, DB implementation, connection, and DB operations are only partially implemented. Includes significant errors that cause the page not to display correctly.	Contains many of the required data layer components (DAO, DB design, DB implementation, connection, DB operations). Code runs with errors, and pages may display with noticeable incorrect layout or behaviour.	Contains at least one of each required data layer component: DAO, DB design, DB implementation, connection, and DB add/delete/update Code runs with minimal or no errors and page layout displays with only minor issues.	Contains required da layer elemer (DAO, I design, DB i plementation, connection, I add/delete/u as well as ex elements. Co runs with errors and pa layout displa correctly.
Deliverable 4: Project Documentation and Video Demonstrations (5 marks)					

Continued on next pa

Component	Absent (F)	Insufficient (1–59.49%)	Emerging (60–69.49%)	Satisfactory (70–76.49%)	Competent Mastering (77–100%)
Documentation & Video	Ideas and evidence are disorganized, making relationships unclear. No or ineffective transitions. Conclusion is absent or simply restates the introduction. Student has an accumulation of errors in grammar, usage, and mechanics that distract from meaning. ERD is not organized, missing statements, contains errors, and has no comments. Video fulfills only some of the project scope with significant requirements missing.	Some but not all of the following: ideas and evidence are sequenced to show relationships; transitions connect ideas; conclusion, when appropriate, follows from and supports the controlling idea; writing is mostly free of distracting errors in grammar, usage, and mechanics. ERD is completed, organized with comments, and without major errors. Video fulfills some significant requirements of the project scope.	Most but not all of the following: ideas and evidence are clearly sequenced to show relationships; transitions connect ideas; conclusion, when appropriate, follows from and supports the controlling idea; writing is free of distracting errors. ERD is completed, organized with comments, and without errors. Video fulfills most significant requirements of the project scope.	All of the following: ideas and evidence are sequenced to show clear relationships; transitions connect ideas; conclusion, when appropriate, follows from and supports the controlling idea; writing is free of distracting errors. ERD is completed, organized with comments, and without errors. Video fulfills all significant requirements of the project scope.	All of the f lowing: ide and evider are logica sequenced show cl relationships; transitions a varied a connect ide showing cl relationships; conclusion, when approp ate, is logi and rai important implications; writing is f of distracti errors. EF is complet organized w comments, a without erro Video co pletely fulf or exceeds t project scope

Individual Grading (50%)

Component	Absent (F)	Insufficient (1–59.49%)	Emerging (60–69.49%)	Satisfactory (70–76.49%)	Mastering (77–100%)
Code Walkthrough (22 marks)					

Continued on next pa

Component	Absent (F)	Insufficient (1–59.49%)	Emerging (60–69.49%)	Satisfactory (70–76.49%)	Mastering (77–100%)
Code Walk-through	Code formatting and coding best practices are not respected. Explains and describes the code structure and the functional and non-functional requirements with limited clarity. Uses language with limited accuracy and effectiveness. Tension and nervousness are obvious and the student has trouble recovering from mistakes.	Some but not all of the following: code formatting and best practices are only partly respected; explains and describes the code structure and requirements with some clarity; uses language with reasonable accuracy and effectiveness; makes mistakes but recovers; displays some tension.	Most but not all of the following: code formatting and coding best practices are respected; explains and describes the code structure and functional and non-functional requirements with considerable clarity; uses language with considerable accuracy and effectiveness; makes minor mistakes but quickly recovers; displays little or no tension.	All of the following: code formatting and coding best practices are respected; explains and describes the code structure and functional and non-functional requirements with considerable clarity; uses language with considerable accuracy and effectiveness; makes minor mistakes but quickly recovers; displays little or no tension.	All of the f lowing: co formatting and codi best practi are respecte explains and c scribes the co structure a functional a non-functiona requirements with a high d gree of clari uses langua with a high d gree of accura and effecti ness; appea relaxed and se confident, wi no noticeal mistakes.
Follow-up Questions and Discussion (28 marks)					
Follow-up Questions & Discussion	Unable to answer questions from the examining board.	Able to answer some but not all questions from the examining board.	Able to answer most but not all questions from the examining board.	Able to answer all questions from the examining board but lacks some confidence.	Able to a swer all qu tions a demonstrate complete und standing of t study, showi confidence in answers.