## *Application Development and Emerging Technologies Final Project*

**1. Choose ONE (1) Mini Application**

Each group must select one (1) from the following mini applications:

1. Student Notes and Reminders Web App
2. Personal To-Do Task Manager
3. Basic Contact Directory Web Application
4. Simple Inventory Tracker for Small Items
5. miniLibrary – Simple Book Borrowing Log

*Option 1: Simple Student Notes and Reminders Web App*

Description: A basic CRUD app where students can write notes, set reminders, and mark items as done.

Core Features:

- Add/Edit/Delete notes
- Mark reminders as completed
- Category filtering (e.g., School, Personal)

Emerging Tech Integration:

- Fetch API (backend → frontend)
- Optional: Simple local notification

Suggested Tech Stack: Node.js, Express.js, MongoDB or JSON file-based storage, Basic HTML/CSS/JS

*Option 2: Personal To-Do Task Manager*

Description: A simple task management web app for listing and tracking tasks.

Core Features:

- Add/Edit/Delete task
- Set task priority (Low, Medium, High)
- Simple user login

Emerging Tech Integration:

- JWT for basic authentication
- Deployed to Vercel/Render

Suggested Tech Stack: Node.js, Express.js, MongoDB, JWT Auth

*Option 3: Basic Contact Directory Web Application*

Description: A CRUD-based address book app that stores contact names, emails, phones, and notes.

Core Features:

- List of contacts
- Add/Edit/Delete contacts
- Search by name

Emerging Tech Integration:

- REST API for contact management
- Optional: Upload small profile icons

Suggested Tech Stack: Node.js, Express.js, MongoDB/MySQL, Basic UI

*Option 4: Simple Inventory Tracker for Small Items*

Description: A lightweight inventory list for tracking household or school items.

Core Features:

- Add items with quantity
- Update quantity
- Delete items
- Low-stock indicator

Emerging Tech Integration:

- API-based CRUD
- Optional: Convert to PWA for offline access

Suggested Tech Stack: Node.js, Express.js, MongoDB, Vanilla JS

*Option 5: miniLibrary – Simple Book Borrowing Log*

Description: A minimal system for recording book check-in/check-out transactions.

Core Features:

- Add books to catalog
- Check out a book
- Return (check-in) a book
- View borrower history

Emerging Tech Integration:

- Basic authentication

- Deployed as small cloud app

Suggested Tech Stack: Node.js, Express.js, MongoDB, JWT or session-based login

**2. Deployment Instructions (FREE Hosting Options)**

Each group must deploy their application online using any of the following free hosting platforms:

Option A: Deploy Backend to Render.com (Free Tier)

Steps:

1. Push your backend folder to GitHub.
2. Go to https://render.com and create an account.
3. Click New → Web Service.
4. Choose your GitHub repository.
5. Set:
    - Build Command:
    - npm install
    - Start Command:
    - node server.js
6. Select Free Tier, then click Deploy.
7. Render will give you a public API URL (e.g., https://yourproject.onrender.com).

Option B: Deploy Full Stack Using Vercel (Frontend + certain Node.js endpoints)

1. Push your frontend folder to GitHub.
2. Go to https://vercel.com.
3. Click New Project → Import GitHub Repo.
4. Configure build settings (if needed).
5. Deploy.
6. Vercel gives you a public URL (e.g., https://yourapp.vercel.app).

Option C: MongoDB Database Hosting (Required if using MongoDB)

Use MongoDB Atlas Free Tier:

1. Go to https://www.mongodb.com/atlas/database
2. Sign up → Create a Free Cluster.
3. Whitelist all IP addresses "0.0.0.0/0".
4. Create a database user and password.
5. Obtain the connection string:
6. mongodb+srv://<username>:<password>@cluster0.mongodb.net/<dbname>
7. Update your Node.js .env file to use this string.

**3. Video Demonstration Requirement**

Each group must submit a **5–10 minute video** containing:

**Introduction**

- Project title
- Group members
- Chosen mini application

**System Demonstration:** Show the **actual working features**:

- Add / Edit / Delete data
- Display list or table
- Login/Logout (if implemented)
- Search or filter features
- Any enhancement your group added

**Deployment Demonstration**

- Show the live deployed URL
- Run the application in the cloud

**GitHub Repository Walkthrough**

- Show commit history
- Branches (if any)
- Final directory structure

**Format Requirements**

- YouTube Unlisted Link
- Clear audio and screen visibility