PITP WEB DEVELOPMENT FINAL TASK ASSIGNMENT
UNIVERSITY WEB PORTAL

Course: Certified Web Developer

Topic: Building a Responsive University Website

Deadline: 8th Jan, 2025

---

1. Assignment Objective

Build a functional, responsive University Portal for **MUET**. You may choose **one** of the following technical tracks to complete this assignment:

- **Track A:** React.js (Component-based, Single Page Application)
- **Track B:** HTML/CSS/JS (Standard Multi-page Website)

Both tracks have equal grading weight. Choose the one you feel most confident in.

---

2. Pages to Build

regardless of the track you choose, your website must contain these **five** distinct sections/pages:

1. **Homepage:** Hero section, Welcome message, Highlights.
2. **About MUET:** History, Mission, Vision, Leadership message.
3. **Admissions:** Eligibility criteria list and Important Dates table.
4. **Departments:** A list of departments with a **Search/Filter** feature.
5. **Contact:** University address and a functional **Inquiry Form** with validation.

---

3. Technical Instructions (Choose Your Track)

Track A: React.js Requirements

- **Framework:** React.js (create-react-app or Vite).
- **Styling:** Bootstrap 5 (React-Bootstrap or standard CDN).
- **State Management:** Use **useState** for creating state variables.
- **Navigation:** Use **React Router** or Implement **Conditional Rendering**.
- **Data Handling:** Create a data.js file with arrays for Departments and display them using .map().
- **Forms:** Build "Controlled Components" where input values are bound to state.

Track B: HTML / CSS / JS Requirements

- **Structure:** Standard HTML5 semantic structure. You may build this as **5 separate HTML files** linked together (home.html, about.html, etc.).
- **Styling:** Bootstrap + a custom style.css file.
- **JavaScript Logic:**
    - ❖ You must link a script.js file at the bottom of your pages.
    - ❖ **DOM Manipulation:** Use document.getElementById or querySelector to interact with elements.
- **Data Handling:** For the Departments page, create a JavaScript array of objects (e.g., const departments = [...]). Use a loop (forEach) to dynamically generate the HTML cards and inject them into the DOM. **Do not hardcode the HTML cards manually.**
- **Forms:** Use addEventListener to handle the form submit event and validate inputs.

---

4. Functional Requirements (Common to Both Tracks)

A. Shared Layout (Header & Footer)

- **Navbar:** Responsive Bootstrap Navbar. Must collapse into a "hamburger menu" on mobile screens.
- **Footer:** Sticky or static footer with Copyright info, Social Links, and Quick Links.

B. Departments Page (The "Logic" Challenge)

- **Data:** Display at least 6 departments (e.g., Software, Civil, Electrical, Mechanical).
- **Search Filter:**
    - ❖ Add an input field at the top of the list.
    - ❖ **Logic:** When the user types in the box, the list of departments must update in real-time to show only matching results.
    - ❖ *(React: Filter the array before rendering / JS: Hide non-matching elements or re-render the list).*

C. Contact Page (The "Validation" Challenge)

- **Form Fields:** Name, Email, Subject, Message.
- **Validation:**
    - ❖ Name cannot be empty.
    - ❖ Email must contain an "@" symbol.
    - ❖ Message must be at least 10 characters long.
- **Feedback:** Display a **Bootstrap Alert** (Green for success, Red for error) based on the validation result.

---

5. Deliverables

1. **Source Code:** Upload your code to a public GitHub repository.
2. **Live Preview:**
    - **React:** Deploy on Vercel.
    - **HTML/JS:** Deploy on GitHub Pages or Vercel.
3. **Documentation:** A README.md file containing:
    - Your Name & Roll Number.
    - Which Track you chose (React or HTML/CSS/JS).
    - Screenshots of the Desktop and Mobile view.

---

6. Submission Format

- **Repository Name:** PITP_WebAssignment_YourName
- **Submit to:** PITP Portal (as instructed).