CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY (CHARUSAT)
FACULTY OF TECHNOLOGY AND ENGINEERING (FTE)
SUBJECT: WEB DEVELOPMENT FRAMEWORKS (ITUE203)
SEMESTER: 3^{RD}, 2025-26 (ODD)
PRACTICAL LIST

Practical Number	Title	CO/PO
1	**Problem Definition:** Initiate the "Project Title" by defining scope, key pages (min. 10), and layout with HTML skeletons. E.g. **StudentHub** • **index.html – Home Page** • **about.html – About Portal** • **register.html – Registration Page** • **login.html – Login Page** • **dashboard.html – Student Dashboard** • **events.html – Events Page** • **profile.html – Student Profile** • **contact.html – Contact Us** • **admin.html – Admin Control Panel** • **faq.html – Frequently Asked Questions** **Key Questions / Analysis / Interpretation:** 1. What pages and features should be included in the student portal? 2. How will navigation and page flow be structured? 3. What are the user roles (e.g., admin, student)? **Supplementary Problems:** Create a sitemap and navigation design. **Key Skills to be addressed:** Requirement analysis, wireframing, HTML5 semantic layout **Applications:** Web application planning, portal design **Learning Outcome:** Students will be able to identify system requirements and create a foundational HTML structure.	CO1
	Dataset/Test Data: N/A (Design and logic only)	
	Tools/Technology To Be Used: VS Code, HTML5, Draw.io/Figma	
	Total Hours: Implementation – 4 hours Total Engagement – 6 hours	
	Post Laboratory Work Description: Documentation of requirements and static HTML layout	
	Evaluation Strategy Including Viva: Wireframe review, role explanation, and page structure analysis Feedback on Problem Definition Implementation (Satisfaction Level 0 to 4, where 0 is lowest,1 is poor,2 is average, 3 is good, 4 is excellent) (This can be asked for group of practical belongs to same tool/concept/technology)	

	Advanced/Intermediate Extension: Intermediate: Create a responsive wireframe using Figma Advanced: Create ER diagram & REST API route plan for backend design	
2	**Problem Definition:** Design a fully responsive layout for the portal home, about, and registration pages using CSS and Flexbox/Grid. **Key Questions:** 1. How does layout change with screen size? 2. Which layout approach is used and why? 3. Are color schemes and fonts readable and user-friendly? **Supplementary Problems:** Theme switcher using CSS variables **Key Skills:** CSS Flexbox, media queries, page layout **Applications:** Multi-device support for UI **Learning Outcome:** Students will design a user-friendly and responsive UI.	CO2
	Dataset/Test Data N/A (UI design only)	
	Tools/Technology: HTML5, CSS3	
	Total Hours: Implementation – 5 hours Total Engagement – 6 hours	
	Post Laboratory Work: Create and test responsive views for key pages	
	Evaluation Strategy: UI responsiveness check and CSS technique analysis **Advanced/Intermediate Extension:** Intermediate: Create 2 additional pages (e.g., Contact, Feedback) Advanced: Convert one page to use templating via JavaScript (Handlebars or JS include)	
3	**Problem Definition:** Create a user registration page with frontend validation using HTML5 and JavaScript. **Key Questions:** 1. Are all input types correctly used? 2. Is JavaScript validation effective and user-friendly? 3. Are errors appropriately handled? **Supplementary Problems:** Password strength meter **Key Skills:** HTML forms, JS form validation **Applications:** Registration, data entry systems **Learning Outcome:** Students will be able to design accessible and validated forms. **Dataset/Test Data:** Sample registration details	CO1, CO3

	Tools/Technology: HTML5, JavaScript (ES6+)	
	Total Hours: Implementation – 5 hours Total Engagement – 6 hours	
	Post Laboratory Work: Submit form with validations, screenshot with test cases	
	Evaluation Strategy: Code inspection, validation demo **Advanced/Intermediate Extension:** Intermediate: Add side navigation menu (hamburger toggle) Advanced: Build responsive layout using Bootstrap or Tailwind CSS	
4	**Problem Definition:** Create dynamic content such as collapsible FAQs, popups, and sliders in portal pages. **Key Questions:** 1. How is the DOM selected and manipulated? 2. Are events and listeners properly handled? 3. How is interactivity enhancing usability? **Supplementary Problems:** Create notification popup banner **Key Skills:** DOM, Event Handling **Applications:** Interactive UIs, dynamic dashboards **Learning Outcome:** Students will apply JavaScript for enhancing user experience.	CO3, CO4
	Dataset/Test Data: Static JSON for events or FAQs	
	Tools/Technology: JavaScript (ES6+), HTML/CSS	
	Total Hours: Implementation – 6 hours Total Engagement – 7 hours	
	Post Laboratory Work: Testing of dynamic modules on different pages	
	Evaluation Strategy: Live demo and source code walkthrough **Advanced/Intermediate Extension:** Intermediate: Add animation transitions to cards/buttons Advanced: Implement CSS theme switcher (light/dark mode)	
5	**Problem Definition:** Display events list and student profiles using object arrays and JSON parsing. **Key Questions:** 1. How is JSON parsed and displayed? 2. What methods are used to manipulate arrays? 3. How is modularity maintained? **Supplementary Problems:** Create pagination logic for JSON data **Key Skills:** Objects, JSON, loops	CO3, CO4

	Applications: Dynamic data rendering **Learning Outcome:** Students will understand modular JS and data handling.	
	Dataset/Test Data: JSON with mock student/event data	
	Tools/Technology: JavaScript, JSON, HTML5	
	Total Hours: Implementation – 6 hours Total Engagement – 8 hours	
	Post Laboratory Work: Integrate JSON with dynamic tables/lists	
	Evaluation Strategy: Console testing and JSON parsing questions **Advanced/Intermediate Extension:** Intermediate: Add country/state select with dependent dropdowns Advanced: Integrate reCAPTCHA or create a custom CAPTCHA using Canvas	
6	**Problem Definition:** Store submitted registration data in a PHP file and confirm submission. **Key Questions:** 1. Are POST/GET methods used correctly? 2. How is data stored and displayed? 3. Are user inputs sanitized? **Supplementary Problems:** Create a simple success/error response page **Key Skills:** PHP POST/GET, input sanitization **Applications:** Form-based applications **Learning Outcome:** Students will create a working backend form processor	CO1, CO5
	Dataset/Test Data: Registration test inputs	
	Tools/Technology: PHP, XAMPP	
	Total Hours: Implementation – 6 hours Total Engagement – 8 hours	
	Post Laboratory Work: Code explanation and XAMPP test case	
	Evaluation Strategy: Form walkthrough and backend validation **Advanced/Intermediate Extension:** Intermediate: Add “Remember Me” functionality with expiration Advanced: Create client-side token system using JWT-like approach	
7	**Problem Definition:** Create a login/logout system with session & cookie handling **Key Questions:** 1. Is the session securely started and terminated?	CO1, CO5

2. Are cookies correctly managed?
3. Is redirection based on login status implemented?

Supplementary Problems:
Remember-me checkbox with cookies

Key Skills:
Sessions, Cookies, Authentication

Applications:
Secure web apps

Learning Outcome:
Students will implement user sessions in PHP

Dataset/Test Data:
Username/password combinations

Tools/Technology:
PHP, Browser dev tools

Total Hours:
Implementation – 6 hours
Total Engagement – 8 hours

Post Laboratory Work:
Login/logout test screenshots

Evaluation Strategy:
Session inspection, cookie check

Advanced/Intermediate Extension:
Intermediate: Implement sorting and filtering of event rows
Advanced: Use external JSON and render dynamically with pagination

8

Problem Definition:
Connect the "StudentHub" portal to a MySQL database to store and retrieve student/event data dynamically. Focus on insert, select, and update operations using PHP and ensure secure DB connectivity and error handling.

CO1, CO5

Sample SQL Dump (Dataset – Table: students):

```
CREATE TABLE students (
student_id INT PRIMARY KEY,
name VARCHAR(50),
email VARCHAR(100),
course VARCHAR(50),
year INT
);
```

student_id	name	email	course	year
1	Riya Sharma	riya@gmail.com	BCA	2
2	Aman Verma	aman.v@gmail.com	B.Sc CS	3
3	Neha Reddy	neha.r@gmail.com	BCA	1
4	Sagar Nair	sagar.n@gmail.com	BSc IT	2
5	Priya Joshi	priya.j@gmail.com	BCA	3

Key Questions:
1. Is the connection established securely?
2. Are insert, select, and update operations working?
3. Is error handling in place?

Supplementary Problems:
Show the latest 5 events on the dashboard using LIMIT

	Key Skills: PHP-MySQL, CRUD, SQL queries **Applications:** Any data-driven system **Learning Outcome:** Students will integrate DB into dynamic sites	
	Dataset/Test Data: SQL Dump (provided)	
	Tools/Technology: MySQL, PHP, phpMyAdmin	
	Total Hours: Implementation – 6 hours Total Engagement – 8 hours	
	Post Laboratory Work: Database dump submission	
	Evaluation Strategy: DB query viva, result output testing	
9	**Problem Definition:** Submit form data using PHP and store it in a text file **Key Questions:** 1. Is the form submitted using POST? 2. Is input validated/sanitized? **3.** Is the confirmation **message displayed?** **Supplementary Problems:** Store in CSV format **Key Skills:** PHP forms, file writing **Applications:** Form processors **Learning Outcome:** Use PHP to collect/store data	CO1, CO5
	Dataset/Test Data: Form inputs	
	Tools/Technology: PHP, XAMPP	
	Total Hours of Implementation: 4 **Total Engagement: 6**	
	Post Lab: Demo file writes	
	Evaluation Strategy: Show form submission trace **Advanced/Intermediate Extension:** Intermediate: Store records in structured format (CSV) Advanced: Store data as JSON file and display it on a webpage dynamically	
10	**Problem Definition:** Develop a login/logout system with session handling. Secure dashboard access should be granted only if the session is active. Use basic PHP session management to store user authentication state. **Sample Users Table:**	CO1, CO5

user_id	username	password	role
1	admin	admin@123	admin
2	student1	stud123	user
3	student2	pass321	user

Key Questions:

1. Are sessions securely started/stopped?
2. Are users redirected after login?
3. Is session persistence maintained?

Supplementary Problems:
Add session timeout

Key Skills:
Sessions, login, redirection

Applications:
Auth backend

Learning Outcome:
Secure user login session

Dataset/Test Data:
Dummy user DB

Tools/Technology:
PHP, XAMPP

Total Hours of Implementation: 5
Total Engagement: 6

Post Lab:
Log in demo with access control

Evaluation Strategy:
Session handling questions

Advanced/Intermediate Extension:
Intermediate: Implement basic role-based access (e.g., admin vs user)
Advanced: Add session timeout or last login tracker

11

Problem Definition:
Implement student data insertion and retrieval from MySQL. Focus on SELECT, INSERT, DELETE. Normalize DB schema and allow search by name.

CO1, CO5

Sample Table: students (same as Practical 8)

Key Questions:

1. Are SQL queries, correct?
2. Are insert, select, and delete working?
3. Is the DB schema normalized?

Supplementary Problems:
Add search by name

Key Skills:
PHP-MySQL, SQL

Applications:
Dynamic DB apps

Learning Outcome:
Create a data-driven page

Dataset/Test Data:
SQL file with students

Tools/Technology:

MySQL, PHP

Total Hours of Implementation: 5
Total Engagement: 6

Post Lab:
SQL dump + UI demo

Evaluation Strategy:
DB output and schema check

Advanced/Intermediate Extension:
Intermediate: Add filter/search functionality on student list
Advanced: Use prepared statements with PDO for secure DB queries

12

Problem Definition:
Develop complete CRUD functionality for managing events. Admin users can add, view, update, and delete events. Reflect changes in the UI dynamically.

CO1, CO5

Sample Table: events

event_id	title	date	location	status
1	Tech Fest	2025-08-10	Seminar Hall	open
2	Hackathon	2025-09-12	Lab Block	closed
3	Coding Marathon	2025-07-22	Online	open
4	Quiz Competition	2025-07-30	Auditorium	open
5	Seminar AI	2025-08-05	Room 202	closed

Key Questions:
1. Are they adding, update, and delete functionalities, correct?
2. Is UI linked with DB correctly?
3. Are success/failure messages shown?

Supplementary Problems:
Add event status (open/closed)
Key Skills:
PHP, MySQL, CRUD
Applications:
Admin tools
Learning Outcome:
Develop a complete CRUD module

Dataset/Test Data:
Events SQL dump

Tools/Technology:
PHP, MySQL

Total Hours of Implementation: 5
Total Engagement: 6

Post Lab:
CRUD demo

Evaluation Strategy:
Code + live test

Advanced/Intermediate Extension:
Intermediate: Add file upload for event posters
Advanced: Use AJAX (Vanilla or jQuery) to perform CRUD without reloading

13

Problem Definition:
Implement validation, sanitization, and password hashing
Key Questions:

CO1, CO3,

	1. Is password_hash() used correctly? 2. Are form inputs validated on both ends? 3. Are SQL injections prevented? **Supplementary Problems:** Add a CAPTCHA field **Key Skills:** Validation, security, SQL **Applications:** Secure user registration/login **Learning Outcome:** Implement secure backend logic	CO5
	Dataset/Test Data: Login form	
	Tools/Technology: PHP, SQL	
	Total Hours of Implementation: 4 **Total Engagement: 5**	
	Post Lab: Secure form submission test	
	Evaluation Strategy: Code inspection + injection tests **Advanced/Intermediate Extension:** Intermediate: Add front-end validation using Regex Advanced: Implement SQL injection prevention and audit logging	
14	**Problem Definition:** Develop an admin dashboard to view/manage users **Key Questions:** 1. Are users listed dynamically from the DB? 2. Are delete/update actions working? 3. Is access restricted to the admin? **Supplementary Problems:** Add user status (active/inactive) **Key Skills:** Admin logic, role-based access **Applications:** Content/user moderation **Learning Outcome:** Build a role-based admin UI	CO1, CO4, CO5
	Dataset/Test Data: DB with multiple users	
	Tools/Technology: PHP, MySQL	
	Total Hours of Implementation: 4 **Total Engagement: 6**	
	Post Lab: Admin demo	
	Evaluation Strategy: Access control validation **Advanced/Intermediate Extension:** Intermediate: Add active/inactive status toggle with DB update Advanced: Use session role management and dynamic menu loading	

15	**Problem Definition:** Integrate all modules into a single deployable “Project Title” Portal **Key Questions:** 1. Are all pages properly linked and navigable? 2. Are sessions and DB working end-to-end? 3. Are validations and security features integrated? **Supplementary Problems:** Add an analytics dashboard **Key Skills:** Full-stack integration **Applications:** Deployable web apps **Learning Outcome:** Deliver a complete, secure web application	CO2, CO4, CO5
	Dataset/Test Data: The entire semester project data	
	Tools/Technology: All technologies used	
	Total Hours of Implementation: 6 **Total Engagement: 8**	
	Post Lab: Full demo + documentation	
	Evaluation Strategy: Holistic viva, performance test **Advanced/Intermediate Extension:** Intermediate: Deploy project locally with Apache Virtual Hosts Advanced: Push project to GitHub and deploy on free hosting (e.g., Render, Vercel with static frontend + backend)	