



Faculty of Engineering and Technology
Department of Computer Science

Course Title:	Web Technologies
Code:	CS-
Coordinating Faculty/ Department:	Faculty of Engineering and Technology department of Computer Science
Credits:	04
Pre-requisites:	Web fundamental, Object-oriented programming
Lecturer Name and Contact:	Khalilullah Akbari +93729908855 k.akbari@kardan.edu.af

GENERAL COURSE INFORMATION

The **Web Technologies** course provides a comprehensive introduction to modern frontend development, focusing on building dynamic and interactive web applications using **JavaScript** and **React.js**. Students will learn core web development concepts, including HTML, CSS, and JavaScript fundamentals, before diving into React to create efficient, scalable, and user-friendly applications. The course covers state management, component-based architecture, responsive design, and best practices for frontend development. By the end of the course, students will have hands-on experience building real-world projects, equipping them with the skills needed to develop modern web applications.

COURSE OBJECTIVES

By the end of this course, students will be able to:

1. **Understand Core Web Technologies** – Gain proficiency in HTML, CSS, and JavaScript to build responsive and interactive web pages.
2. **Master JavaScript Fundamentals** – Learn modern JavaScript (ES6+) concepts, including functions, objects, arrays, and asynchronous programming.
3. **Develop Interactive User Interfaces with React.js** – Build scalable and maintainable web applications using React.js and its component-based architecture.
4. **Manage State Efficiently** – Utilize React's state management techniques, including hooks and context API, for dynamic user experiences.
5. **Implement Responsive Design** – Design mobile-friendly and accessible web applications using CSS frameworks and best practices.
6. **Work with APIs** – Fetch and handle data from RESTful APIs to create real-time, data-driven applications.

7. **Build and Deploy Web Applications** – Develop a complete frontend project and deploy it using modern tools and hosting platforms.

This course equips students with the skills needed to confidently develop modern, high-performance web applications using JavaScript and React.js.

COURSE LEARNING OUTCOMES

This course is designed to provide a comprehensive introduction to both front-end web development, equipping students with the skills needed to build modern, responsive, and dynamic web applications. The course is structured as follows:

Introduction to Web Development:

Overview of the web development process and the key technologies involved.

Understanding the roles of front-end, back-end, and full-stack development.

HTML & CSS:

Basics of HTML for structuring web content.

CSS for styling and layout, including responsive design techniques.

Introduction to CSS frameworks like Bootstrap for faster development.

JavaScript Fundamentals:

Core JavaScript concepts, including variables, functions, and control structures.

DOM manipulation and event handling to create interactive web pages.

Introduction to modern JavaScript (ES6+), including arrow functions, promises, and async/await.

Advanced JavaScript & Front-End Frameworks:

Working with APIs to fetch and display data.

Introduction to JavaScript frameworks like React or Vue.js for building single-page applications (SPAs).

State management and component-based architecture in modern web apps.

Introduction to Back-End Development:

Project Development & Deployment:

Capstone project where students build a fully functional web application.

Version control with Git and collaboration using GitHub.

Focus on security, including input validation and protection against common vulnerabilities.

Performance optimization techniques for faster web applications.

Introduction to testing and debugging tools for ensuring application quality.

LEARNING REFERENCES

Following resources will be used as references during teaching this course:

- John Ducket: JavaScript and JQuery
- Wieruch, R. (2022). *The road to React: Your journey to master React.js in JavaScript* (5th ed.). Independently published.

TEACHING METHODS

Following teaching methods will be adopted for the taught course:

1. Lecture:
2. Presentation:
3. Small talk
4. Group projects
5. Practical work
6. Experiential Techniques:
7. Simulations

CLASSROOM ETIQUETTE:

Class room courtesy is expected of all students all the time. Following is general guideline for the class room conduct:

1. Refrain from chatting with fellow students.
2. Arrive on time. If you come late please be courteous to other students
3. Do not get up and walk out halfway through class.
4. Do not cut the instructor off at the end of class.
5. All cell phones and pagers must be turned off during class and examination.
6. Eating is not permitted.
7. Above all please be cognizant of the learning process and purpose of you being in the
8. Class room and extend same respect to other students.

POLICIES / GUIDELINES

Guidelines for Format of Assignments: (General Recommendations)

1. Assignments must be submitted online using Kardan LMS
2. No Assignment will be accepted after the deadline
3. Hard copies of the assignment won't be accepted
4. Assignments via e-mail won't be accepted
5. Assignments need to be written by hands also students are obliged to upload the screen shots of the assignments
6. Copied assignments will lead to the cancellations of the assignments of the respective students
7. Proper references relevant to the assignment must be included in the last page of the assignment
8. Students are also advised to include the table of contents in assignment

Note: All assignments and any work related to this work submitted will be subjected to plagiarism verification by software.

COURSE SCHEDULE / CONTENTS

Week No.	Topics for Discussion	Book Chapter #	Suggested Activity	Topic Outcomes
1	JavaScript Fundamentals • JavaScript role in frontend development • Variables and constants • Data types • Operators • Conditional statements • Loops • Arrays and basic array methods	From lecture notes	Create a program to solve a real-world problem.	Will learn about course
2	Functions and Objects • Function declaration/expression • Arrow functions • Parameters and return values • Scope • Introduction to closures • Objects and object properties • Destructuring • Spread/rest operators	From lecture notes	Code-along: Interactive to-do list.	Students will be able to implement JS object and JS BOM in their projects
3	DOM Manipulation • DOM concept and structure • Selecting elements with <code>querySelector()</code> / <code>querySelectorAll()</code> • Changing text and HTML • Attributes and classes • Creating and removing elements • Basic Browser APIs	From lecture notes	Implementation of DOM and form validation	Students will be able to implement DOM and form validation in their projects
4	Events and Form Validation • Mouse events • Keyboard events • Form events • <code>addEventListener()</code> • Event bubbling • Event delegation • Basic form validation	From lecture notes	API-based weather app.	Students will learn events and animation in the javascript
5	Modern JavaScript and Browser Storage • Template literals • Optional chaining		Building a simple React app.	React structure and create simple react app.

	• Nullish coalescing • Modules (import / export) • JSON • Local Storage and Session Storage			
6	Asynchronous JavaScript • Synchronous vs asynchronous execution • Callbacks • Promises • async/await • Fetch API • Basic error handling	Chapter 3	Build a counter app with props and state	Understanding state and props and code using them
7	React, JSX and Component Basic • Understanding JSX • Components • Building functional and class components • Props and their validation	Chapter 5	Refactor an old app using hooks.	Build a timer app using hooks.
8	React Hooks • useState, • useEffect, • custom hook	Chapter 10	Build tax calculator, another page calculator, another page money exchange	Create a multi-page blog app
9	React Router and Forms • SPA concept • React Router • Routes and navigation • Route parameters • Controlled forms • Form validation	From Lecture notes	Build a shopping cart using Context API.	Understand Redux and context API
10	Shared State and Component Communication • Parent/child communication • Lifting state • Context API • useContext() • Introduction to global state • When global state is needed	Chapter 12	Create a basic server.	Implement a CLI tool using Node.js
11	Advanced React Concepts • Context API and Redux (state management). • Higher-order components and performance optimization.	Chapter 17	Build website for small e-commerce company	Implement company website

12	REST API Integration • REST concepts • GET, POST, PUT/PATCH, DELETE • Fetch/Axios • Loading states • Error states • CRUD frontend	From class lecture	Build a database for a task manager	Integrate a database into an existing API
13	Advanced Topics • Code splitting and lazy loading • Error boundaries	From class lecture	Implement JWT authentication	Authentication functionality task.
14	React Application Organization and Performance • Project folder structure • Reusable components • Custom hooks • Code splitting concept • Lazy loading • React.memo() concept - useMemo, and useCallback	From class lecture	Deploy a full-stack app.	Deployed project.
15	Deployment and Introduction to Modern Front-End Practices • Production build • Environment variables • Git/GitHub basics • Deployment with Vercel or similar platform • Introduction to Next.js concepts	From class lecture	Midpoint project review.	Will be able to work in project
16	Final Project Presentation and Review • Project demonstration • Architecture explanation • Technical Q&A • Problems encountered and solutions • Course review	From class lecture	Final project grading.	Q&A

COURSE ASSESSMENT

TYPE	PERCENTAGE	RATIONALE
------	------------	-----------

Internal Assessment Assignments: 10% Quizzes: 10% Attendance: 5%	25 %	Equal weightage is assignments and quizzes and at the same time emphasizing on the importance of class participation.
External Assessment Midterm: 25% Terminal: 50%	75%	Students develop an examination sense through midterm examination hence 25% weightage is appropriate. Midterm duly followed by terminal examination providing 50% weightage which is an opportunity to improve scores appropriately.

GRADING

MARK RANGE	GRADE POINT	GRADE	EXPECTED RESULT
90 and above	4.0	A	At least 25% expected to secure A grade
80 – 89	3.0	B	At least 35% expected to secure B grade
70 – 79	2.0	C	At least 30% expected to secure C grade
60 – 69	1.0	D	At least 7% expected to secure D grade
59 and below	0.0	F	At least 3% expected to secure F grade

KARDAN UNIVERSITY POLICY ON PLAGIARISM:

All examinations and quizzes will be “closed book” unless otherwise instructed. At the time of examination all students are requested to clear their desks and are not allowed exchanging any notes or electronic (text) messages to other students. All cellular phones should be in silent mode and student will not be allowed to use it during the examination other than medical/family/work emergency. All students are expected to adhere to these policies and procedures.

CONDUCT AND IMPORTANT POLCIES:

Any student found guilty of a breach of ethics will be referred to Disciplinary Committee of the University.

- Breach of ethics includes, but is not limited to plagiarism (the copying of other’s ideas and passing them off as one’s own); copying or other forms of cheating on examinations, papers, and reports; the sale, purchase, or distribution of term papers It is within an instructor’s discretion to impose a lesser penalty, e.g., “zero” grade on a given assignment.
- Course registration is charged by the management. Please approach the management for any queries about course enrolment. In no circumstances should you approach the lecturers who have no control on this.
- Make-up exam for midterm and terminal exam is available only for those individuals, who are not able to attend their regural exams. Provision of supported documents are mandatory for grant of approval to participate

in make-up exams. Those students who miss their regular exam without a genuine reason, will be entitled for 80% of total makeup marks.

Usually make-up exam starts a week after the regular exam finishes.

d) There is no make-up session for the oral presentations and quizzes. If you are absent from the oral presentation/ quiz without eligible reasons/documents, you will not earn any marks.

e) All examinations and quizzes will be “closed book” unless otherwise instructed. At the time of examination all students are requested to clear their desks and are not allowed exchanging any notes or electronic (text) messages to other students. All cellular phones should be in silent mode and student will not be allowed to use it during the examination other than medical/family/work emergency. All students are expected to adhere to these policies and procedures.

f) There is no supplementary exam for any failed course. Individuals, who fail the course, must retake the module.

ATTENDANCE:

Your regular and punctual attendance at lectures and seminars is expected in this course.

University regulations indicate that if students attend less than 65% of scheduled classes they may be refused final assessment.

This course policy has been approved in the Curriculum Development Committee meeting FCS/CDC-2026-03.

Head of Committee Signature.....

Dean, Faculty of Computer Science Signature