

Frontend Development with React Course Policies & Syllabi, 2024

Course Name	Frontend Development with React
Duration of the Course in Weeks	26 weeks
Duration of the Course in Academic Hours	156 (of which 60% practice hours)
Tuition Fee	600,000 Kazakhstani Tenge (100,000 KZT per month)
Learning Format	Online – pre-recorded lectures + live practice hours (Google Meet)
Learning Platform	In-house LMS + Google Meet
Frequency of Classes	Weekly lectures + bi-weekly practice (combined 6 hours)
Skills / Results Upon Completion of Training	Proficiency in building dynamic user interfaces using React
Qualification Upon Completion of Training	Certificate of Completion in "Frontend Development with React"
Methodological Basis of the Course	Active Learning Hands-on practice
The Ratio of Theory to Practice	60-40 (60% practice, 40% theory)
Teaching Staff	Sufyan Mustafa bin Uzayr
Tracking the Development of Students	Assessment and Evaluation Feedback Mechanisms Project Milestones Coding Portfolios Peer and Self-Assessment
Student Performance Evaluation Methods	Midterms + Coding Tests + Final Exam

Course Description

This course offers a comprehensive exploration of front-end web development using React, a powerful library for building interactive user interfaces. Students will master React fundamentals including JSX syntax, component-based architecture, and state management. Advanced topics such as React Router, Redux, and responsive design principles will be covered, empowering students to develop dynamic and responsive web applications. Through hands-on projects and assignments, participants will gain practical experience in building real-world React applications, equipped with modern development tools and best practices. Upon completion, students will possess the skills and knowledge necessary to excel in front-end development with React in today's competitive web development landscape.

Course Duration

Number of weeks	26
Number of academic hours	156 (of which 60% practice hours)

Pre-requisites

- As the course materials, lectures, and instructions are primarily delivered in English, a strong understanding of the English language is essential for effective comprehension and communication.
- Fundamental logical and reasoning abilities are required to understand code concepts and solve problems efficiently.

Skills Taught

- Proficiency in React fundamentals, including JSX syntax, component structure, and state management.

- Understanding of advanced React concepts such as React Router for client-side routing and Redux for centralized state management.
- Application of responsive design principles to ensure optimal user experience across different devices and screen sizes.
- Integration of third-party libraries and APIs into React applications.
- Utilization of modern development tools and workflows, including npm or Yarn for package management and Webpack for build automation.
- Debugging techniques using browser developer tools to identify and resolve frontend issues efficiently.
- Ability to structure and organize React applications effectively, employing best practices for modularity and reusability.

Learning Plan

Week	Topic	Notes
1	HTML Fundamentals. Introduction to HTML and its role in web development. Understanding HTML tags, elements, and attributes. Creating basic HTML documents and structuring web pages. Working with text, links, images, and tables in HTML. HTML forms and input elements for user interaction.	
2	CSS Fundamentals. Introduction to CSS and its role in styling web pages. Understanding CSS syntax, selectors, and properties. Applying styles to HTML elements using inline, internal, and external CSS. Working with colors, fonts, backgrounds, and borders in CSS. CSS layout techniques including positioning, floats, and flexbox.	
3	JavaScript Basics. Introduction to JavaScript and its role in web development. Understanding	

	JavaScript syntax, variables, data types, and operators. Working with control structures like loops and conditional statements. Manipulating the DOM (Document Object Model) with JavaScript. Handling events and user interactions using JavaScript event listeners.	
4	jQuery Introduction. Introduction to jQuery library and its benefits for web development. Selecting and manipulating HTML elements with jQuery Working with jQuery event handling and animations. Implementing AJAX (Asynchronous JavaScript and XML) with jQuery for dynamic content loading. Building interactive web components using jQuery UI widgets.	
5	Introduction to React Fundamentals. Introduction to React library and its advantages. Understanding JSX syntax for writing React components. Creating and rendering React components.	
6	Managing State and Props in React. Understanding state and props in React applications. Managing component state and updating UI accordingly. Passing data between components using props.	CT-1
7	Exploring Component Lifecycle Methods. Understanding the lifecycle of React components. Implementing componentDidMount, componentDidUpdate, and componentWillUnmount lifecycle methods. Handling component updates efficiently.	
8	Conditional Rendering and Handling Lists in React. Implementing conditional rendering based on component state. Rendering lists of data	

	dynamically using map method. Understanding the importance of keys in React lists.	
9	Introduction to React Hooks. Understanding the motivation behind React Hooks. Exploring useState and useEffect hooks for managing state and side effects. Refactoring class components to functional components with hooks.	
10	Implementing Forms and Controlled Components in React. Creating forms and handling form submissions in React. Implementing controlled components to manage form state. Validating form input and displaying error messages.	
11	Setting Up React Router For Navigation. Installing and configuring React Router for client-side routing. Defining routes and implementing navigation between different views in a React application.	
12	Creating Routes And Nested Routes In React Applications. Setting up nested routes for more complex navigation structures. Implementing route parameters to handle dynamic URL segments.	
13	Midterm-1	Midterm-1
14	Implementing Route Parameters And Query Strings. Passing and accessing route parameters in React Router. Handling query strings for additional data transmission in URLs.	
15	Protecting Routes With Authentication Using React Router. Implementing authentication guards to restrict access to certain routes. Redirecting users based on their authentication status using React Router.	

16	Introduction To Redux And Its Core Principles Understanding the Redux state management pattern and its principles. Exploring the concepts of actions, reducers, and the single source of truth. To Redux and its Core Principles	
17	Setting Up Redux In A React Application Installing and configuring Redux in a React application. Setting up the Redux store and connecting it to React components.	
18	Defining Actions And Reducers In Redux Creating Redux actions to describe state changes. Defining reducers to specify how state should change in response to actions.	CT-2
19	Integrating Redux With React Components. Connecting React components to the Redux store using the connect function. Accessing Redux state and dispatching actions from React components.	
20	Asynchronous Actions And Middleware In Redux Handling asynchronous operations in Redux using middleware like Redux Thunk or Redux Saga. Implementing async actions to fetch data from APIs and update the Redux store.	
21	Data Visualization with Matplotlib. Making HTTP requests to fetch data from external APIs in a React application. Handling response data and updating component state with fetched data.	
22	Fetching Data from APIs Using Fetch or Axios Managing loading states and error handling when fetching data asynchronously. Implementing best practices for efficient data fetching in React applications.	
23	Handling Asynchronous Data Fetching in React	

	Applications. Installing and integrating UI libraries like Material-UI or React Bootstrap into React projects. Utilizing pre-built components and styles to enhance the UI of React applications.	
24	Library integration, data caching/pagination, responsive design (Flexbox/CSS Grid), React performance/accessibility optimization.	
25	End Term (Midterm-2)	Midterm-2
26	Final Project: Dynamic Web Application Development with React	
Finals Period		Final Exam

Student Evaluation Criteria

Component	Description	When	Points
Class Test	(a) Computer-based (b) Online	6th and 18th week	each worth 15 points → 30 points
Midterms	(a) Computer-based (b) Online	13th and 25th week	each worth 15 points → 30 points
Final Exam	(a) Computer-based (b) Online	Finals Period	40 points
Total			100 points

Letter Grade	Points	Grade Remarks
--------------	--------	---------------

A	95-100	Excellent
A-	90-94	
B+	85-89	Good
B	80-84	
B-	75-79	
C+	70-74	
C	65-69	Satisfactory
C-	60-64	
D+	55-59	
D	50-54	
FX	25-49	Fail
F	0-24	

Course Fee

600,000 Kazakhstani Tenge (100,000 KZT per month)

Teaching Staff

Sufyan Mustafa bin Uzayr
Telegram Username - @sufyanbinuzayr
Site: https://sufyanism.com/
Google Scholar Profile - https://scholar.google.com/citations?user=CzF2dJIAAAJ

Sufyan Mustafa bin Uzayr is a skilled teacher with extensive experience in the IT industry, having taught for over 5 years. He is proficient in various software tools and programming paradigms. Having authored 60+ books, their expertise in Sufyan has taught and worked at universities and colleges in 3 different continents.

Student Selection Criteria

To be admitted to the program, students are required to pass an Aptitude Test, consisting of 25 questions. The Test shall be conducted online, the dates of which shall be communicated to the students via email.

Additionally, the Test topics shall include General Reasoning and Logic, Mathematical Skills (School Grade-8 level).

It is a pre-requisite for students to possess general verbal and oral English skills.

Training Format

Online – pre-recorded lectures + live practice hours (Google Meet)

Use of Technology in Teaching

Zeba Academy uses its own flagship Learning Management System, that comes with features such as Pre-recorded Video Lessons, Advanced Quizzes, Flashcards, Image Hotspots Groups and Cohorts, AI-based adaptive learning algorithm, and guided practice materials in the form of ebooks and other content.

Contact

Zeba Academy Help Desk for Students

Telegram and WhatsApp: **+7 705 435 2915**