LAB & SKILL WORKBOOK

23SDCS12A / 23SDCS12E / 23SDCS12R
FULL STACK APPLICATION DEVELOPMENT

TEAM FSAD
K L UNIVERSITY | VADDESWARAM

LAB & SKILL WORKBOOK

23SDCS12A / 23SDCS12E / 23SDCS12R

FULL STACK APPLICATION DEVELOPMENT

STUDENT NAME	
STUDENT ID	
YEAR	
SEMESTER	
SECTION	
FACULTY NAME	

DEPARTMENT VISION AND MISSION

Vision

To be a Department of International Repute through Continuous Research, Innovation and Industry Led Curriculum.

Mission

To Impart Quality Education with Social Consciousness and make them globally competent.

Program Educational Objectives

1. Practice engineering in a broad range of industrial, societal and real-world applications.
2. Pursue advanced education, research and development, by adapting creative and innovative practices in their professional careers.
3. Conduct themselves in a responsible, professional, and ethical manner.
4. Participate as leaders in their fields of expertise and in activities that support service and economic development throughout the world.

PROGRAM OUTCOMES		
PO	**Graduate Attributes**	**Program Outcome Description**
1	**Engineering Knowledge**	To impart mathematics, science, & engineering knowledgeto develop skills to solve complex engineering problems.
2	**Problem Analysis**	Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3	**Design/ development of solutions**	Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4	**Conduct investigations of complexproblems**	An ability to use research-based knowledge and research methods including design of experiments, analysis and interpretation of data and synthesis of the information to provide valid conclusions.
5	**Modern tool usage**	Ability to create, select and apply appropriate techniques, resources and modern engineering activities, while understanding its limitations.
6	**The engineer and society**	Ability to apply reasoning and the contextual knowledge toassess social & health, safety, legal and cultural issues and the consequent responsibilities relevant to the professionalengineering practices.
7	**Environment and sustainability**	Ability to demonstrate the engineering knowledge to find solutions to contemporary issues by understanding their impact on societal and environmental contexts, towards sustainable development
8	**Ethics**	An ability to apply ethical principles and commit to professional ethics and responsibilities and norms of engineering practice.
9	**Individual and teamwork**	To inculcate abilities to be able to act as a leader as well as team player effectively in multi-disciplinary settings
10	**Communication**	To develop oral and written communication skills to articulate the complex engineering activities with the engineering community and and society effectively through reports and design documentation, make effective presentations, and give and receive clear instructions.
11	**Project management and finance**	To develop working knowledge and understanding of the engineering and management principles to manage projects in multi-disciplinary environments.
12	**Lifelong learning**	To inculcate the habit of constant knowledge upgrading habit to meet the ever-changing technology and industry needs.
PROGRAM SPECIFIC OUTCOMES		
PSO1	An ability to design and develop software projects as well as to analyze and test user requirements.	
PSO2	Working knowledge on emerging technologies as per the industry requirements	

LAB EXPERIMENTS

2024-25 EVEN SEMESTER FSAD LAB CONTINUOUS EVALUATION

Lab No.	Date of Evaluation	Implementation (20M)	Output (20M)	Viva Voce (10M)	Total (50M)	Faculty Signature
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						

2024-25 EVEN SEMESTER FSAD SKILL CONTINUOUS EVALUATION

Team Details

Team ID	Section	Student ID	Student Name	Team Lead

Problem Statement ID:

Problem Statement:

Project Title:

Innovative / Unique Ideology for the Application:

Git Repo Link:

2024-25 EVEN SEMESTER FSAD SKILL CONTINUOUS EVALUATION

Review 1 - Rubrics and Evaluation

S.no	Review 1 - Rubrics	0	5	7	9	10	Marks
1	Survey Conducted	No work done	Survey conducted but partial	Survey conducted is satisfied	Extensive survey conducted through online	Extensive survey conducted including field survey	
2	Survey Summary	No work done	Survey summary is partial	Survey summary is satisfied	Unique points identified from summary	Excellent feature or measure is identified from survey summary to application yet to develop	
3	Design Thinking Concepts (Empathy Mapping, Persona, Customer Journay Map, etc...)	No work done	Design thinking concepts are partial	Design thinking concepts are satisfied	Design thinking concepts are good	Design thinking concepts are excellent	
4	Innovative and Unique Add-on to the Application	No work done	Innovative idealogy is in partial state	Identified innovative / unique points for application	Good deviation from existing app	Excellent innovation	
5	Module Identification (Sample: user, admin, authentication, core functionality modules, result display module, feedback module, information module, etc..)	No work done	Modules are partial	Requirements are satisfied for the project	Requirements are identified in good level	All the requirements are completed	
6	Article Publishing	No work done	Partially published	Article is satisfactory	Article is good	Article is excellent	
7	Prototype Creation	No work done	Prototype is partial	Prototype is satisfactory	Prototype is good	Prototype is excellent	
8	Youtube Video on Explaining the Prototype	No work done	Video prepared in offline and yet to upload	NA	NA	Video published	
9	Team Coordination	No work done	Individual student's coordination with team is just ok	Individual student's coordination with team is satisfactory	Individual student's coordination with team is good	Individual student's coordination with team is excellent	
10	Overall Project Progress	No work done	There is progress but not enough	Project progress is satisfactory	Project progress is good	Project progress is excellent	

Signature with Faculty ID

2024-25 EVEN SEMESTER FSAD SKILL CONTINUOUS EVALUATION

Review 2 - Rubrics and Evaluation

S.No	Review 2 - Rubrics	0	5	7	9	10	Marks
1	Front End Elements / Components in React + Vite App	No work done	Required elements are partially brought inside the app	70% elements and components are made	All components are developed in front end	Components segragation and availability is excellent	
2	Responsive Front End Design (CSS) in React + Vite App	No work done	Design with CSS is done but partiall	Design completed but not responsive	Design is good and responsive	Design feels and look excellent	
3	Front End Event Handling in React + Vite App	No work done	Event handling is partially completed	Event handling is satisfied	Event handling is good	Event handling is excellent	
4	Git Repo for Team Work	No work done	Just created and not utilized	Git push and pull operations are satisfactory	Git push and pull operations are good	Git utilization among team members is excellent	
5	Team Coordination	No work done	Individual student's coordination with team is just ok	Individual student's coordination with team is satisfactory	Individual student's coordination with team is good	Individual student's coordination with team is excellent	
6	Overall Project Progress	No work done	There is progress but not enough	Project progress is satisfactory	Project progress is good	Project progress is excellent	

Signature with Faculty ID

2024-25 EVEN SEMESTER FSAD SKILL CONTINUOUS EVALUATION

Review 3 - Rubrics and Evaluation

S.No	Review 3 - Rubrics	0	5	7	9	10	Marks
1	Responsive Front End in React + Vite App (including Multiple Components, CSS, Event Handling)	No work done	Design is done but partiall	Design completed but not responsive	Design is good and responsive	Design feels and look excellent	
2	Server Side Integration	No work done	Basic Server Running	All dependencies and configurations setted	Server handles request and response	Server Integration is good with react + vite app	
3	Modules Development	No work done	<30% modules in back end completed	30% modules in back end completed	40% modules in back end completed	50% modules in back end completed	
4	Database Usage	No work done	Connections Established	Entity classes created	JPA integrated	Data storing in database	
5	Authentication Process Integrating with Server	No work done	Authentication tried to integrate with server	Only signup page integrated and working	signup and login integrated with server and working	signup and login integrated with server and operations are perfect	
6	Team Coordination	No work done	Individual student's coordination with team is just ok	Individual student's coordination with team is satisfactory	Individual student's coordination with team is good	Individual student's coordination with team is excellent	
7	Overall Project Progress	No work done	There is progress but not enough	Project progress is satisfactory	Project progress is good	Project progress is excellent	

Signature with Faculty ID

2024-25 EVEN SEMESTER FSAD SKILL CONTINUOUS EVALUATION
Review 4 - Rubrics and Evaluation

S.No	Review 4 - Rubrics	0	5	7	9	10	Marks
1	Responsive Front End in React + Vite App (including Multiple Components, CSS, Event Handling)	No work done	Design is done but partiall	Design completed but not responsive	Design is good and responsive	Design feels and look excellent	
2	Module Implementation	No work done	<70% implementation completed	70% implementation completed	90% implementation completed	All modules perfectly implmented	
3	Additional Features (like - encryption & decrypt, jwt tokens, spring cloud, mail, google maps, search / filter module, payment gateway, file storing in database, Captcha Generation, etc...)	No work done	Additional Features Implementation is just ok	Additional Features Implementation is satisfying	Additional Features Implementation is good	Additional Features Implementation is excelent	
4	Database Usage	No work done	Connections Established	Data storing in database	The database tables, fields and operations are good enough	The database tables, fields and operations are excellent	
5	Role Based Access	No work done	Role based access is partial	Role based access is satisfactory	Role based access is good	Role based access is excellent	
6	Team Coordination	No work done	Individual student's coordination with team is just ok	Individual student's coordination with team is satisfactory	Individual student's coordination with team is good	Individual student's coordination with team is excellent	
7	Overall Project Progress	No work done	There is progress but not enough	Project progress is satisfactory	Project progress is good	Project progress is excellent	

Signature with Faculty ID

2024-25 EVEN SEMESTER FSAD SKILL CONTINUOUS EVALUATION
Review 5 - Rubrics and Evaluation

S.No	Reviews 5 - Rubrics	0	5	7	9	10	Marks
1	Responsive Front End in React + Vite App (including Multiple Components, CSS, Event Handling)	No work done	Design not satisfactory	Design completed but not responsive	Design is good and responsive	Design feels and look excellent	
2	Module Implementation	No work done	<80% implementation completed	80% implementation completed	90% implementation completed	All modules perfectly implmented	
3	Additional Features (encryption & decrypt, jwt tokens, spring cloud, mail, google maps, search / filter module, payment gateway, file storing in database, Captcha Generation, etc…)	No work done	Additional Features Implementation is just ok	Additional Features Implementation is satisfying	Additional Features Implementation is good	Additional Features Implementation is excelent	
4	FSAD - Advanced - Microservies, load balancing, api gateway. FSAD - Regular - Unique Feature Implementation	No work done	ADV - Microservices server page is present / REG - Component available for unique feature	features are implemented in a satisfactory level	features are implemented in a good level	features are implemented in a excelent level	
5	Deployment Done	No work done	Deployment is partially working	Front end deployed	Frontend and Backend deploymed with database	Deployed with CI/CD	
6	Project Git Integration	No work done	Git repo is upto date and number of operations over the git is ok	Git repo is upto date and number of operations over the git is satisfactory	Git repo is upto date and number of operations over the git is good	Git repo is upto date and number of operations over the git is excellent	
7	Youtube Video on Explaining the Project	No work done	Video prepared in offline and yet to upload	NA	NA	Video published	
8	Team Coordination	No work done	Individual student's coordination with team is just ok	Individual student's coordination with team is satisfactory	Individual student's coordination with team is good	Individual student's coordination with team is excellent	
9	Overall Project Progress	No work done	There is progress but not enough	Project progress is satisfactory	Project progress is good	Project progress is excellent	

Signature with Faculty ID

DEPARTMENT OF CS & IT
COURSE CODE: 23SDCS12A / 23SDCS12E / 23SDCS12R
FULL STACK APPLICATION DEVELOPMENT

Date of the Session: __/__/____ **Time of The Session:_____to_____**

LAB – 1 ➔ Implementing grid, flex and block display

Prerequisites:

Knowledge on the HTML elements and styling

Exercise:

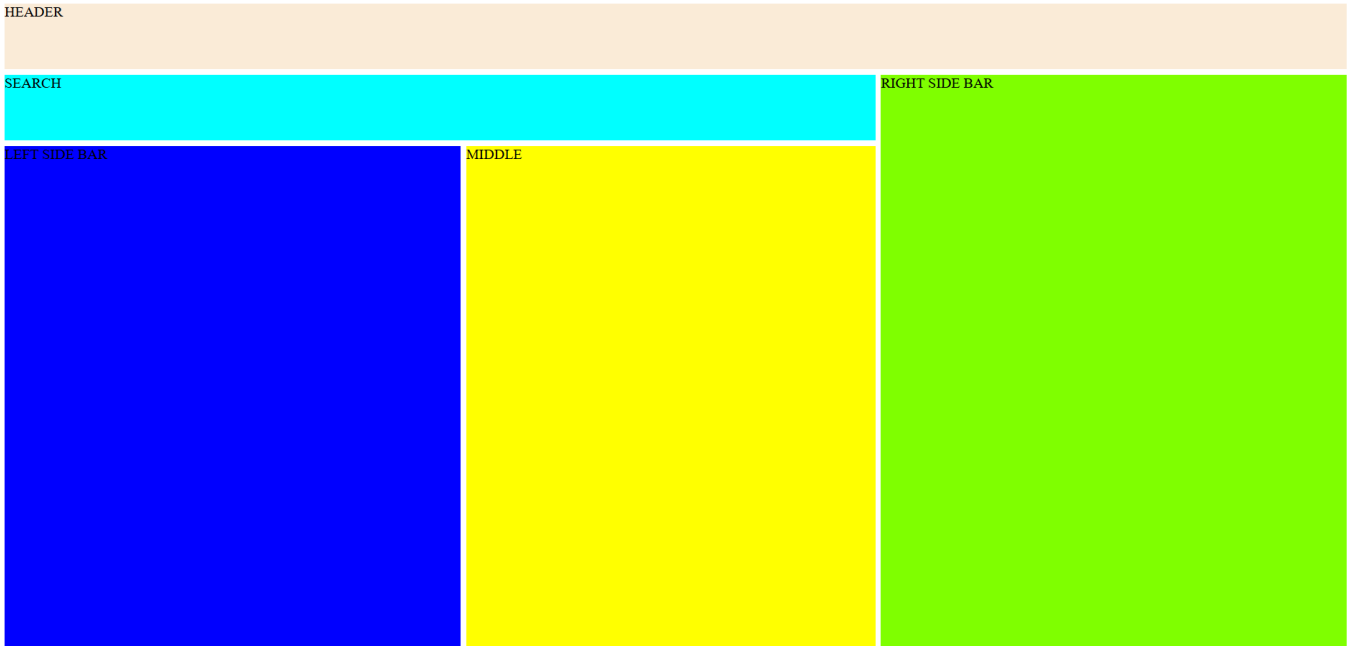
Faculty and students are organizing a "Greeks for Greeks" student chapter focused on teaching practical web development concepts. As part of this event, they plan to demonstrate a project that illustrates how to build a **product grid** using **CSS Grid**, and individual **product cards and sidebar** using **Flex and block display** model in React. Each component will be styled using external CSS to create a cohesive and responsive layout. How can they set up this example to effectively showcase these layout techniques in a React application?

Project structure:

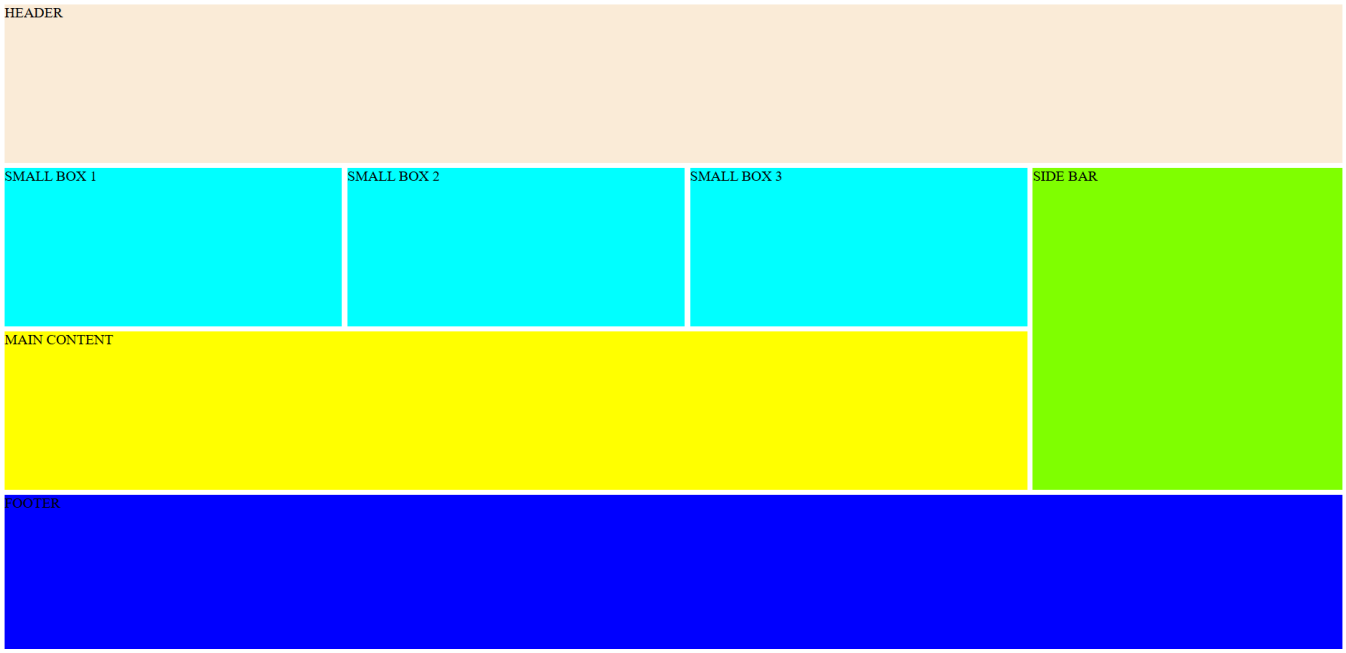
```
product-catalog/
├── src/
    ├── App.jsx
    ├── Grid.jsx
    ├── Sidebar.jsx
    ├── MainContent.jsx
    └── Main.jsx
```

Layout should be one of the below,

Sample Layout 1



Sample Layout 2



VIVA QUESTIONS:

1. What is the difference between display: grid, display: flex, and display: block?
2. Explain the CSS Grid layout. How does it help in building complex layouts?
3. What is the grid-template-columns and grid-template-rows property in CSS Grid? How do you use them?
4. How does Flexbox simplify layout management in CSS?
5. What is the difference between justify-content in Flexbox and CSS Grid?

(For Evaluator's use only)

Comment of the Evaluator (if Any)	Evaluator's Observation Marks Secured:_____out of 50 Evaluator Emp ID: Evaluator Signature with Date

DEPARTMENT OF CS & IT
COURSE CODE: 23SDCS12A / 23SDCS12E / 23SDCS12R
FULL STACK APPLICATION DEVELOPMENT

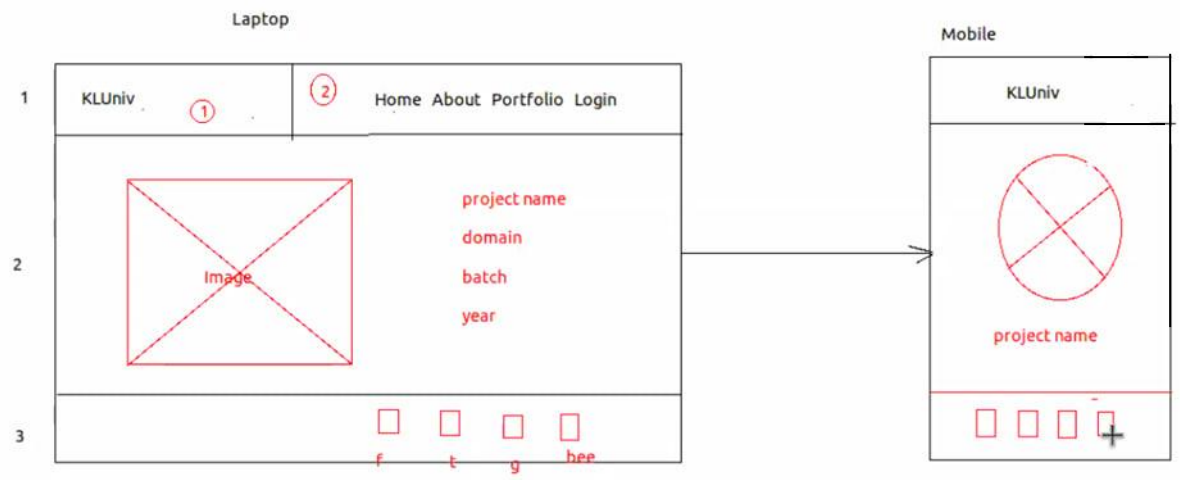
Date of the Session: __/__/____ **Time of The Session:_____to_____**

LAB – 2 ➔ Development of responsive frontend for system and mobile View

Prerequisites:
Knowledge on the HTML elements and styling

Exercise:
This Product Directory allows users to view product details in a grid layout on larger screens and in a single-column list view on smaller screens. It will dynamically fetch employee data from the database and adjust the layout based on the screen size. So that, to create a **responsive front end** that display data and adapts to both **desktop** and **mobile** views.

The view may be like the below one,



VIVA QUESTIONS:

1. How does the className attribute work in JSX, and why isn't it just class like in HTML?
2. What are dynamic classes, and how can you add them based on state in React?
3. How do CSS preprocessors like Sass or LESS integrate with React?
4. Explain the concept of scoped CSS in React and how you can achieve it.
5. What are styled-components, and how do they differ from inline styles or external stylesheets?

(For Evaluator's use only)

Comment of the Evaluator (if Any)	Evaluator's Observation Marks Secured:_____out of 50 Evaluator Emp ID: Evaluator Signature with Date

DEPARTMENT OF CS & IT
COURSE CODE: 23SDCS12A / 23SDCS12E / 23SDCS12R
FULL STACK APPLICATION DEVELOPMENT

Date of the Session: __/__/____ **Time of The Session:_____to_____**

LAB – 3 ➔ Working with React props, state

Prerequisites:
Knowledge on the HTML elements and styling

Knowledge on the Java Script

Exercise

The TCS team lead need visited to KLU and they need User Profile Component as a real-time example, where we manage the user's personal information (like name, age, and location) in a parent component (App.js) and pass that data to a child component (UserProfile.js) using props. The user can also update this information, and we will manage the updates using state within the child component.finally TCS team need the following pages.

Pass user data (name, age, and location) to a child component as props.

Allow the user to update their profile, which will be handled via state in the child component. The updated profile information will be reflected back in the parent component using state lifting (passing the updated state back to the parent).

```
user-profile/
├── src/
  ├── App.js
  ├── UserProfile.js
  └── index.js
```

VIVA QUESTIONS:

1. What are props in React, and how are they different from state?
2. How would you pass props from a parent component to a child component? Provide an example.
3. Explain how state is managed in React and how it differs from props.
4. What is the purpose of this.setState in class components?
5. How can you update state in functional components using hooks?

(For Evaluator's use only)

Comment of the Evaluator (if Any)	Evaluator's Observation Marks Secured:_____out of 50 Evaluator Emp ID: Evaluator Signature with Date

DEPARTMENT OF CS & IT
COURSE CODE: 23SDCS12A / 23SDCS12E / 23SDCS12R
FULL STACK APPLICATION DEVELOPMENT

Date of the Session: __/__/____ **Time of The Session:_____to_____**

LAB – 4 ➔ Implementation of multiple components as a "Single Page React App" with Redux State Management for Routing.

Prerequisites:
Having a basic understanding of HTML and CSS is valuable for building user interfaces and styling your React components.

>npm install redux

Exercise 1:

Create a single page simple React application with two routes: Home and About. Implement navigation links to switch between these routes using Redux.

Exercise 2:

Implement a nested routing structure in a React application using Redux. Create a parent route and two child routes that are rendered within the parent component. All these as a single page app.

VIVA QUESTIONS:

1. How does Redux help in managing the state of an application?
2. List the key components of Redux and their roles?
3. What is the role of reducer and store in any e-commerce application.
4. What are the benefits of using Redux in terms of debugging and maintaining application state?
5. What is drawback developer need to face without redux?

(For Evaluator's use only)

Comment of the Evaluator (if Any)	Evaluator's Observation Marks Secured:_____out of 50 Evaluator Emp ID: Evaluator Signature with Date

DEPARTMENT OF CS & IT
COURSE CODE: 23SDCS12A / 23SDCS12E / 23SDCS12R
FULL STACK APPLICATION DEVELOPMENT

Date of the Session: __/__/____ **Time of The Session:_____to_____**

#LAB – 5 ➔ Utilizing the predefined responsive design elements with Bootstrap / Material-UI for designing

Prerequisites:
Material UI need to be installed or Bootstrap CDN need to be integrated

Exercise 1:

Create a form with Material-UI / Bootstrap components (such as text fields, select fields, and checkboxes) to collect user information and validate the input.

Exercise 2:

Implement a responsive navigation bar using Material-UI's App Bar / Bootstrap component. The navigation bar should have a logo, menu items, and handle mobile responsiveness.

VIVA QUESTIONS:

1. How do you add Material-UI to a React project?
2. Describe the Box component in Material-UI and its use cases.
3. How do you add Bootstrap to a React project?
4. What are the main differences between Bootstrap and Material-UI?
5. What is React-Bootstrap, and how does it differ from regular Bootstrap?

(For Evaluator's use only)

Comment of the Evaluator (if Any)	Evaluator's Observation Marks Secured:_____out of 50 Evaluator Emp ID: Evaluator Signature with Date

DEPARTMENT OF CS & IT
COURSE CODE: 23SDCS12A / 23SDCS12E / 23SDCS12R
FULL STACK APPLICATION DEVELOPMENT

Date of the Session: __/__/____ **Time of The Session:_____to_____**

LAB – 6 ➔ Transferring (Sending and Receiving) data with Axios / Fetch API in React

Prerequisites:

Axios / Fetch need to be installed

Exercise 1:

Create a React component called UserData that fetches user data from a given API endpoint using Axios. Display the fetched user data in a table format.

Use this API end point to get user data - https://api.github.com/users/defunkt/following or https://reqres.in/api/users

Exercise 2:

Send API request to 3rd party and get the response to print it.

Use this API call - https://rapidapi.com.

For example weather api (use any api as your wish) -https://rapidapi.com/worldapi/api/open-weather13/playground/apiendpoint_d15cd885-e8e5-49e7-b94b-588c41687aa1

VIVA QUESTIONS:

1. How do you handle loading, success, and error states in a React component when making an API request with Axios or Fetch?
2. Explain how you would use useEffect with Axios or Fetch to make API calls in a React functional component. What are the potential issues, and how would you prevent unnecessary re-renders?
3. How can you pass headers or authentication tokens with Axios or Fetch in a React application, and what's the best way to keep this information secure?
4. How would you cancel an API request in React if the component unmounts before the request completes, using Axios or Fetch?
5. How do you handle asynchronous operations with Axios / Fetch?

(For Evaluator's use only)

Comment of the Evaluator (if Any)	Evaluator's Observation Marks Secured:_____out of 50 Evaluator Emp ID: Evaluator Signature with Date

DEPARTMENT OF CS & IT
COURSE CODE: 23SDCS12A / 23SDCS12E / 23SDCS12R
FULL STACK APPLICATION DEVELOPMENT

Date of the Session: __/__/____ **Time of The Session:_____to_____**

LAB – 7 ➔ Spring Boot Web MVC Demo & Annotations

Prerequisites:

General Idea on Spring Boot MVC Architecture

Exercise:

1. Create a Spring Boot Web MVC application that demonstrates various request mappings and handling methods using a customer object. Implement the following demo operations
2. Create a method that maps to the URL "/demo1" and returns an integer.
3. Create a method that maps to the URL "/demo2" and returns a double value
4. Create a method that maps to the URL "/demo3" using @GetMapping and returns a formatted HTML string.
5. Create a method that maps to the URL "/demo4" using @GetMapping, initializes a string "KLEF", and returns a concatenated string "I Study at " with the initialized string.
6. Create a method that maps to the URL "/demo5/{id}" using @GetMapping, accepts a path variable "id", and returns its value.
7. Create a method that maps to the URL "/demo6/{a}/{b}" using @GetMapping, accepts two path variables "a" and "b", and returns their sum as a string.
8. Create a method that maps to the URL "/demo7" using @GetMapping, accepts a request parameter "id", and returns it as a string.
9. Create a method that maps to the URL "/demo8/{name}" using @GetMapping, accepts a path variable "name", and returns it as a string.
10. Create a method that maps to the URL "/addcustomer" using @PostMapping, accepts a Customer object in the request body, adds it to a list, and returns a confirmation message "Customer Added Successfully".
11. Create a method that maps to the URL "/viewcustomer" using @GetMapping and returns the list of all Customer objects added.

VIVA QUESTIONS:

1. What are the key differences between Spring MVC and Spring Boot MVC?
2. How does Spring Boot simplify the development of Spring applications?
3. Explain how to use @RequestBody and @ResponseBody annotations in Spring Boot.
4. Explain the role of the @Autowired annotation in Spring Boot and how it works in the context of Dependency Injection?
5. What is the purpose of application.properties or application.yml in a Spring Boot application?

(For Evaluator's use only)

Comment of the Evaluator (if Any)	Evaluator's Observation Marks Secured:_____out of 50 Evaluator Emp ID: Evaluator Signature with Date

DEPARTMENT OF CS & IT
COURSE CODE: 23SDCS12A / 23SDCS12E / 23SDCS12R
FULL STACK APPLICATION DEVELOPMENT

Date of the Session: __/__/____ **Time of The Session:_____to_____**

LAB – 8 ➔ Spring Boot with Rest API and CRUD Operations

Prerequisites:
General Idea on Spring Boot MVC and Form Handling
General Idea on Spring Data JPA

Exercise:
Develop a Spring Boot web application to manage a list of products in a warehouse. The application should handle CRUD operations to manage product details such as Product ID, Name, Description, Price, and Quantity. The application should include features to add new products, display a list of all products, update existing product details, and delete products from the database. Use Spring Web MVC for handling HTTP requests, Spring Data JPA for database interactions. Ensure the application is configured to connect to a MySQL/PostgreSQL database and implement both setter-based or constructor-based dependency injections to manage service and repository layers effectively.

VIVA QUESTIONS:

1. Can you explain the role of each layer (Controller, Service, Repository) in a Spring Boot MVC application, especially in the context of CRUD operations?
2. How would you configure and connect a Spring Boot application to a relational database, and what dependencies are necessary for CRUD operations?
3. Describe how you would create and map a JPA entity for a table in the database. How does this mapping support CRUD operations?
4. How do you handle data validation in a Spring Boot CRUD application before saving data to the database? Can you give examples of annotations used for validation?
5. What is the purpose of @Transactional in Spring Boot, and how does it ensure data consistency during CRUD operations?

(For Evaluator's use only)

Comment of the Evaluator (if Any)	Evaluator's Observation Marks Secured:_____out of 50 Evaluator Emp ID: Evaluator Signature with Date

DEPARTMENT OF CS & IT
COURSE CODE: 23SDCS12A / 23SDCS12E / 23SDCS12R
FULL STACK APPLICATION DEVELOPMENT

Date of the Session: __/__/____ **Time of The Session:_____to_____**

LAB - 9 ➔ Spring Boot with ReactJS Integration

Prerequisites:

Implementation skill in Reactjs

Implementation skill in Spring Boot Application with JPA and Database

Exercise:

Implement the necessary ReactJS front end pages for sending the request and receive responses from the back end (spring boot app) designed as per the below requirements.

Develop a Spring Boot web application to manage a list of products in a warehouse. The application should handle CRUD operations to manage product details such as Product ID, Name, Description, Price, and Quantity. The application should include features to add new products, display a list of all products, update existing product details, and delete products from the database. Use Spring Web MVC for handling HTTP requests, Spring Data JPA for database interactions. Ensure the application is configured to connect to a MySQL/PostgreSQL database and implement both setter-based or constructor-based dependency injections to manage service and repository layers effectively.

VIVA QUESTIONS:

1. How does data flow between a ReactJS frontend and a Spring Boot backend in a full-stack application?
2. How do you configure CORS in a Spring Boot application to allow requests from a ReactJS frontend?
3. What is the role of JPA in a Spring Boot application, and how does it interact with a database?
4. How do you handle asynchronous operations in React when fetching data from the Spring Boot API?
5. Can you describe a typical CRUD operation cycle from the ReactJS frontend to the database via Spring Boot and JPA?

(For Evaluator's use only)

Comment of the Evaluator (if Any)	Evaluator's Observation Marks Secured:_____out of 50 Evaluator Emp ID: Evaluator Signature with Date

DEPARTMENT OF CS & IT
COURSE CODE: 23SDCS12A / 23SDCS12E / 23SDCS12R
FULL STACK APPLICATION DEVELOPMENT

Date of the Session: __/__/____ **Time of The Session:_____to_____**

LAB - 10 ➔ Implementing Authentication and Role Based Access

Prerequisites:

Implementation skill in Reactjs

Implementation skill in Spring Boot Application with JPA and Database

Exercise:

Implement the necessary ReactJS front end pages (signin, signup, home, admin) for sending the request and receiving the responses from the back end (spring boot app) designed to handle signin and signup request with database integration.

During registering the user, get input as admin or user. This is to store the user role in database. Have logout button in the front end to sign-out the authentication.

While login, permit the user or admin to Home page on successful login. Permit the admin to admin page and not the user.

VIVA QUESTIONS:

1. How do you implement authentication in a ReactJS and Spring Boot application, and what are the typical steps involved?
2. How do you implement role-based access control (RBAC) in a ReactJS application integrated with a Spring Boot backend?
3. How do you handle session management and token expiration in ReactJS when dealing with authenticated requests?

(For Evaluator's use only)

Comment of the Evaluator (if Any)	Evaluator's Observation Marks Secured:_____out of 50 Evaluator Emp ID: Evaluator Signature with Date

DEPARTMENT OF CS & IT
COURSE CODE: 23SDCS12A / 23SDCS12E / 23SDCS12R
FULL STACK APPLICATION DEVELOPMENT

Date of the Session: __/__/____ **Time of The Session:_____to_____**

#LAB – 11 ➔ Implementing JWT Tokens with encryption and decryption

Prerequisites:

Basic Idea on Spring Security

Basic Idea on JWT Tokens & RBAC

Exercise:

Develop a Spring Boot application with JWT-based security for role-based authentication and authorization.

Call the JWT token generation function and validation function from browser to generate and validate the tokens.

Encrypt the provided data to the token and then where required decrypt the token to get the original data to display in the browser.

VIVA QUESTIONS:

1. What is JWT (JSON Web Token)?
2. How does JWT facilitate secure authentication between a React frontend and a Spring Boot backend?
3. How do you create and configure security filters for handling JWT tokens in Spring Boot?
4. Discuss the use of @PreAuthorize and @Secured annotations in role-based access control.

(For Evaluator's use only)

Comment of the Evaluator (if Any)	Evaluator's Observation Marks Secured:_____out of 50 Name of the Evaluator: Evaluator Signature with Date

DEPARTMENT OF CS & IT
COURSE CODE: 23SDCS12A / 23SDCS12E / 23SDCS12R
FULL STACK APPLICATION DEVELOPMENT

Date of the Session: __/__/____ **Time of The Session:_____to_____**

LAB - 12 ➔ Implementing Microservices and Load Balancing

Prerequisites:
Implementation skill on Spring Boot App
Basic Idea on Microservices and Load Balancing

Exercise 1:

Now there is a demand to make some mathematical calculation over web. So, you need to create one server application (eureka) and two client applications, in which "client 2" will do the calculation and return the answer to "Client 1". The client 1 will handle the user interaction by getting input from user after then sending the same to "client 2" and then getting answer from "client 2" to return the same to "client 1". The server need to monitor both the clients.

Exercise 2:

Now the number of requests from the end users is raising above the threshold level, so its time to create "client 3" which is similar to "client 2" and have load balancer between them. Attach the load balancer in "client 1", so that, every request come to "client 1" need to be load balanced between "client 2" and "client 3".

VIVA QUESTIONS:

1. How can we access RESTful in Microservices?
2. What's the difference between a microservices-oriented architecture (MOA) and a service-oriented architecture (SOA)?
3. What does the term "bounded context" mean in relation to microservices?
4. Explain three types of Tests for Microservices?
5. How is distributed tracing used in Microservices?

(For Evaluator's use only)

Comment of the Evaluator (if Any)	Evaluator's Observation Marks Secured:_____out of 50 Evaluator Emp ID: Evaluator Signature with Date

DEPARTMENT OF CS & IT
COURSE CODE: 23SDCS12A / 23SDCS12E / 23SDCS12R
FULL STACK APPLICATION DEVELOPMENT

Date of the Session: __/__/____ **Time of The Session:_____to_____**

#LAB - 13 ➔ Implementing Spring Cloud Integration

Prerequisites:

Basic Idea on Spring Boot

Basic Idea on Spring Cloud

Exercise:

Now, due to the implementation of microservices in market, there is a requirement to have common values for the same variables available in different spring boot app. In this case, there should be similar 2 spring boot app which uses same variable and the value for the variable need to be picked form the github repo based on the “development” or “production” environment. Assign the picked value to the variable and print it to show the output.

VIVA QUESTIONS:

1. What is Spring Cloud, and why is it useful in a microservices architecture?
2. What is a Circuit Breaker in Spring Cloud, and how does it help improve system resilience?
3. How do you implement centralized configuration management with Spring Cloud Config?
4. What is Spring Cloud Gateway, and how does it differ from Zuul as an API Gateway?
5. How can you secure microservices in Spring Cloud, and what role does Spring Security play?

(For Evaluator's use only)

Comment of the Evaluator (if Any)	Evaluator's Observation Marks Secured:_____out of 50 Evaluator Emp ID: Evaluator Signature with Date

DEPARTMENT OF CS & IT
COURSE CODE: 23SDCS12A / 23SDCS12E / 23SDCS12R
FULL STACK APPLICATION DEVELOPMENT

Date of the Session: __/__/____ **Time of The Session:_____to_____**

#LAB - 14 ➔ Hosting backend (spring boot)

Prerequisites:

Implementation skill on Spring Boot App

Exercise:

Now, it time to bring the web application in online. Whatever you had developed as a spring boot application (full stack) with JPA, database and frontend interaction. You need to host the back end (spring boot) and mysql database in cloud. So that everyone should be able to open this anywhere in the world.

VIVA QUESTIONS:

1. What is Spring Boot, and how does it simplify the Spring framework?
2. Explain the purpose of the @SpringBootApplication annotation.
3. How does Spring Boot handle dependency management, and what is the role of the starter dependencies?
4. What is an embedded server, and which servers are supported by Spring Boot?
5. Explain the difference between @RestController and @Controller.

(For Evaluator's use only)

Comment of the Evaluator (if Any)	Evaluator's Observation Marks Secured:_____out of 50 Evaluator Emp ID: Evaluator Signature with Date

DEPARTMENT OF CS & IT
COURSE CODE: 23SDCS12A / 23SDCS12E / 23SDCS12R
FULL STACK APPLICATION DEVELOPMENT

Date of the Session: __/__/____ **Time of The Session:_____to_____**

#LAB - 15 ➔ Hosting frontend (ReactJS)

Prerequisites:

Implementation skill on ReactJS App

Exercise:

Now, it time to bring the web application in online. What ever you had developed as a ReactJS application (full stack) with back end interaction. You need to host the front end (ReactJS) in cloud. So that every one should able to open this any where in the world.

VIVA QUESTIONS:

1. What is ReactJS, and how does it differ from other JavaScript frameworks?
2. Explain the Virtual DOM and how it improves performance in React.
3. What are functional and class components? When would you choose one over the other?
4. What are React Hooks, and why were they introduced?
5. Describe the purpose of useState and useEffect hooks.

(For Evaluator's use only)

Comment of the Evaluator (if Any)	Evaluator's Observation Marks Secured:_____out of 50 Evaluator Emp ID: Evaluator Signature with Date