

FULL STACK DEVELOPMENT				
Course Code	21CSA763		Credits	2
Hours/Week (L-T-P-S)	1-0-2		CIE Marks	50
Total Teaching Hours	26		SEE Marks	50
Exam Hours	03		Course Type	AEC
Course Component	AEC			
COURSE LEARNING OUTCOMES				
Students will be able to:				
<div><div>1.</div><div>Develop the React applications using states and components trees.</div></div> <div><div>2.</div><div>Build Flexible, Loosely Coupled and Highly scalable Software Applications using spring core.</div></div> <div><div>3.</div><div>Design and apply the concepts of spring MVC in the web application Development.</div></div> <div><div>4.</div><div>Use of tools and practices adopted at industry to build the Software at high velocity using spring boot.</div></div> <div><div>5.</div><div>Analyse the concepts of DevOps and implement the Use-case walkthrough</div></div>				
COURSE CONTENTS				
UNIT -1- (5 Hrs)				
<b>React Introduction</b> - Welcome to React, Redact’s Past and Future, Working with the Files <b>JavaScript for React</b> - Declaring Variables, Creating Functions, Compiling JavaScript, Asynchronous JavaScript, Classes, ES6 Module <b>Pure React</b> - Page Setup, The Virtual DOM, React Elements, React DOM, Children, Constructing Elements with Data, React Components, DOM Rendering, Factories <b>React with JSX</b> - React Elements as JSX, Babel, Recipes as JSX, Intro to Webpack <b>Props, State, and the Component Tree</b> - Property Validation, Refs, React State Management, State Within the Component Tree				
UNIT -2- (5 Hrs)				
<b>Spring Core</b> <b>Spring Core Introduction / Overview</b> - Shortcomings of Java EE and the Need for Loose Coupling, Managing Beans, The Spring Container, Inversion of Control, The Factory Pattern, <b>Configuration Metadata</b> - XML, @Component, Auto-Detecting Beans, Dependencies and Dependency Injection (DI) with the Bean Factory, Setter Injection <b>Spring Container</b> - The Spring Managed Bean Lifecycle, Auto wiring Dependencies <b>Dependency Injection</b> - Using the Application Context, Constructor Injection, Factory Method, Crucial Namespaces ‘p’ and ‘c’, Configuring Collections				
UNIT -3- (5 Hrs)				
<b>Metadata / Configuration</b> - Annotation Configuration @Autowired, @Required, @Resource, @Component, Component Scans. Component Filters, Life Cycle Annotations, Java Configuration, @Configuration, XML free configuration, The Annotation Config Application Context <b>Developing Web applications with Spring MVC</b> <b>RESTful Web Services</b> -Core REST concepts, REST support in Spring 5.x, Use Spring MVC to create RESTful Web services, REST specific Annotations in Spring, Working with Rest Template, URI Templates, @PathVariable, @RequestParam, JSON and XML data exchange, @RequestMapping				
UNIT -4- (5 Hrs)				
<b>Spring Boot</b> <b>SPRING BOOT Introduction</b> - Spring rebooted, Getting started with Spring Boot, developing your first Spring Boot application, Component scans, Auto Configuration, Externalizing configuration with properties, Application Context, Configuring Logging, Spring Boot Dependencies <b>Spring Data REST</b> - Introduction & Overview, Adding Spring Data REST to a Spring Boot Project, Configuring Spring Data REST, Repository resources, Default Status Codes, Http methods, Spring Data REST Associations, Define Querv method				

### UNIT -5- (6 Hrs)

**Introduction to DevOps** - What is DevOps, Evolution of DevOps, Agile Methodology, Why DevOps, Agile vs DevOps, DevOps Principles, DevOps Lifecycle, DevOps Tools, Benefits of DevOps, Continuous Integration and Delivery pipeline, Use-case walkthrough.

GitHub - Introduction to Git, Version control, Repositories and Branches, Working Locally with GIT, Working Remotely with GIT.

Why DevOps, Agile vs DevOps, DevOps Principles, DevOps Lifecycle, DevOps Tools, Benefits of DevOps, Continuous Integration and Delivery pipeline, Use-case walkthrough

**GitHub** - Introduction to Git, Version control, Repositories and Branches, Working Locally with GIT, Working Remotely with GIT.

### TEXTBOOKS

SINO	Unit	Textbook Title	Author(s)	Publisher(s)	Edition/Year of Publication
1	All	Design Patterns Elements of Reusable Object-Oriented Software	Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides	Addison-Wesley Professional	Latest
2	All	Professional Java Development with the Spring Framework	Craig Walls	Wiley	2005
3	All	Spring in Action	Craig Walls	Manning Publications	5th edition
4	All	Spring Boot in Action	Craig Walls	Manning Publications	5th edition
5	All	Learning React: Modern Patterns for Developing React Apps	Book by Alex Banks and Eve Porcello	O'Reilly	Second
6	All	Learning React: Functional Web Development with React and Redux	Book by Alex Banks and Eve Porcello	O'Reilly	First
7	All	DevOps Tools from Practitioner's Viewpoint	Deepak Gaikwad and Viral Thakkar	Wiley	Latest

### ONLINE RESOURCES

#### Links

<https://www.coursera.org/learn/design-patterns>

<https://www.react.express/jsx>

<https://spring.io/projects/spring-framework>

<https://devops.com/>

### COURSE ASSESSMENT METHOD

#### Continuous Internal Evaluation (CIE):

Programs writing ,Execution ,Viva 15

Course project 20

Test 15

Total 50

#### Semester End Examination (SEE):

Final Examination will be conducted 50 Marks.

### PEDAGOGY

- Black Board/PowerPoint Presentations
- Demonstration of Applications using IDE and Development Tools
- Lab Practices

LABORATORY EXERCISES	
SINO	LIST OF EXPERIMENTS-PART A
1	Design a web counter application for the scenario discussed below. Consider that a class has 30 students, and the percentage of each student has also been given. Count the number of students who have cleared the previous exam with distinction (85% and above) and first class (60% to 84%). This information is required by the admin department to analyze how much scholarship needs to be released by college based on the count.
2	Design a React JS application using forms. The application should contain a text area and the input text values should change when the state is updated. When the user changes the state, the application should print the updated value on the browser screen.
3	Write a react application for Validating an input while it's receiving characters is another great application for use Effect. While the input is being stored in a state using useState, the validation takes place every time the input changes, giving immediate feedback to the user
4	Dependency Injection: Design a java code for Injecting dependencies into classes, making them loosely coupled and easier to test.
5	Bean Configuration and Management: Construct a bean configuration and Management program for defining and managing beans, including their lifecycle, scopes, and dependencies.
6	a) Develop a simple web application that allows users to view and manage a list of books. b) Create the Book class. c) Create the BookController class to handle requests and manage the list of books d) Create the views using the Thymeleaf template engine. Assuming the templates are located in the resources/templates directory: e) add-book.html f) Configure the Spring MVC application using Java-based configuration. g) Create the main class to run the application.
	<b>Part B</b> Students have to submit course project

CO-PO-PSO MAPPING															
CO	PO												PSO1	PSO2	PSO3
	1	2	3	4	5	6	7	8	9	10	11	12			
1	2	2	3	-	3	-	-	-	1	1	-	-	2	3	-
2	2	2	3	-	3	-	-	-	1	1	-	-	2	3	-
3	2	2	3	-	3	-	-	-	1	1	-	-	2	3	-
4	2	2	3	-	3	-	-	-	1	1	-	-	2	3	-
5	3	2	3	-	-	-	-	-	3	3	-	-	2	3	-
CL	2	2	3	-	3	-	-	-	1	1	-	-	2	3	-