



Birla Institute of Technology & Science, Pilani
Work Integrated Learning Programmes Division

Digital Learning Handout

Part A: Content Design

Course Title	Full Stack Application Development
Course No(s)	SE ZG503
Credit Units	4
Credit Model	1-1-2 (32 Hours of Class-room Instruction + 32 Hours of Case-studies/Tutorials/Laboratories + 64 Hours of Student Preparation)
Instructors	Akshaya Ganesan
Version No:	2
Date:	17/01/2025

Course Description:

Evolution in web app architectures: Client Server - 2 tier, 3/n tiered, Layered; Distributed - SOA, Web Services, Microservices, Cloud (IaaS/PaaS/FaaS); Modern application landscape; Web applications: Typical structure of end-to-end application; Application components-Frontend / Backend / API / Database / Services; Web Browsers - Client WebAPIs - Browsers APIs for storage, audio, video; Web Assembly; Responsive web; Web Servers; Load balancers; Application servers; API gateways; ORM; DNS; HTTP/S: HTTP headers, HTTP messages, HTTP request methods; Caching; Modern application architectures and Tech-Stacks: Microservices, Serverless; Application development Stacks-Conventional; Modern JavaScript stacks; (Full-Stack) Application Development: Languages (client/server side), Frameworks; Platforms, Deployment (on-prem/cloud); Databases(RDBMS/NoSQL); Interactions(method calls, APIs/REST, messaging); MEAN/MERN as exemplar frameworks.

Course Objectives

No	Course Objective
C01	Understand the modern application landscape and the evolution of the application landscape.
C02	Build an understanding of an end-to-end application's typical structure, design, and implementation considerations.





C03	Comprehend the necessity and usefulness of the client and server-side frameworks, along with their strengths and weaknesses.
C04	Develop and test a working model of web application using the tech stack.

Text Book(s):

T1	Web Development with Node and Express by Ethan Brown , Oreilly Media 2nd Edition , 2019
T2	RESTful Web APIs by Leonard Richardson and Mike Amundsen , Oreilly Media, 2013

Reference Book(s) & other resources:

R1	Full-Stack React Projects by Shama Hoque, Second Edition, ,2020, Packt Publishing
R2	The Design of Web APIs by Arnaud Lauret Published by Manning Publications; 1st edition (November 2019)
R3	Full Stack Web Development: The Comprehensive Guide by Philip Ackermann Shroff/Rheinwerk Computing; First Edition (2 August 2023)
R4	Microservice APIs: Using Python, Flask, FastAPI, OpenAPI and More. Peralta, J. H. (2023): Manning.
R5	GRPC: Up and Running: Building Cloud Native Applications with Go and Java for Docker and Kubernetes Book by Danesh Kuruppu and Kasun Indrasir, Oreilly Media
Web Resources	Mozilla Developer Network https://developer.mozilla.org/en-US/ React JS https://react.dev/learn

Content Structure

Module 1: Application Development

- Introduction to the various Application Landscape: [Web applications, Mobile applications, Cross Platform applications, Cloud native applications, Serverless Applications]
- Layered Architectures (client /server, 2/3 tier- N tier)
- Monolithic
- Distributed Architectures – Service-oriented architecture, Microservices
- MVC Pattern

Module 2: Understanding the Basics

- Structure of web applications
 - Frontend (HTML, CSS, JavaScript),





- Backend Server Side logic, API, Web Services
- Database
- Client – Server Communication
- Relationship between URLs, Domains, and IP Addresses
- Domain Name Systems, Content Delivery Networks
- Technologies and Tools for Full stack development

Module 3: Web Protocols

- HTTP
 - HTTP Request- Response and its structure
 - HTTP Methods;
 - HTTP Headers
 - Connection management - HTTP/1.1 and HTTP/2
- Synchronous and asynchronous communication
- Communication with Backend
 - AJAX, Fetch API
 - Webhooks
 - Server-Sent Events
- Polling
- Bidirectional communication - Web sockets

Module 4: Server Side: Implementing Web Services

- REST
 - Principles of REST
 - REST constraints
 - Service Design with REST
 - Interaction Design with HTTP
 - Interface Design (URI)
 - Representation and Metadata design
 - Implementing REST API
 - Using a Framework
 - URL Mapping
 - Routing Requests-Redirection
 - Implementing a web server
 - Processing request, response, data
 - Storing data in databases





- Models
 - Object Relational Mapper
 - Interaction with DBs
- API versioning and documentation
- GraphQL
 - Schemas and Types
 - Thinking in Graphs
 - Serving over HTTP
 - Implementing the GraphQL API
 - Validation and Execution
- gRPC
 - Service Definition- Protobuf
 - Architecture
 - Channels Streaming and Types

Module 5: Securing Application

- Basic Authentication
- API Authorization
- JSON Web Tokens
- OAuth
- SSL, TLS and HTTPS
- Common Vulnerabilities

Module 6: Understanding Frontend Development

- Designing and Structuring Webpages
- Making pages Interactive with JavaScript
- Using Browser based Web APIs- DOM, Web Storage
- Client-side JavaScript Frameworks: Features and Advantage, MEAN, MERN
- Implementing Single Page Applications using JavaScript Tech stack
 - The Node Ecosystem
 - Project Setup
 - Creating and styling components
 - Managing Component hierarchies and lifecycle





- Managing State
- Routing
- Building, deploying and Hosting

Module 7: Testing

- API Testing
 - API Testing and Types
 - Unit Testing
 - Contract Testing
- Frontend
 - Unit testing
 - Cross browser testing
 - Acceptance Testing

Module 8: Accessibility and Performance

- Accessibility
 - Inclusive Design
 - Assistive Technologies
 - Web content accessibility Guidelines
 - Optimizing Websites for Accessibility
 - Testing Accessibility
- Performance
 - Tools and Metrics for Measuring Performance
 - Options for Optimization
 - Caching- Client side, Server side
 - Minifying Code, Compressing files
 - Lazy Loading

Module 9: Latest Advancements

- Progressive Web Apps
- Web Assembly
- Microfrontends

Learning Outcomes: Students will be able to

LO1	Understand the underlying architecture used for Web applications and identify the various components of the Web Application
-----	---





LO2	Demonstrate the creation of API to accomplish various backend functionalities of an application like database interaction, handling user requests
LO3	Design and develop user-friendly and interactive web frontends.
LO4	Implement a functional end-to-end web application using client-side and server-side web technologies.

Part B: Learning Plan

Contact Session	List of Topic Title	Sub-Topics	Reference
1	Module 1: Application Development	Module 1: Application Development <ul style="list-style-type: none"> • Introduction to the various Application Landscape: [Web applications, Mobile applications, Cross Platform applications, Cloud native applications, Serverless Applications] • Layered Architectures (client /server, 2/3 tier- N tier) • Monolithic • Distributed Architectures – Service-oriented architecture, Microservices • MVC Pattern 	R3- Chapter 12, R1 Web references
2	Module 2: Understanding the Basics	Module 2: Understanding the Basics <ul style="list-style-type: none"> • Structure of web applications <ul style="list-style-type: none"> ▪ Frontend (HTML, CSS, JavaScript), ▪ Backend Server Side logic, API, Web Services ▪ Database • Client – Server Communication • Relationship between URLs, Domains, and IP Addresses • Domain Name Systems, Content Delivery Networks • Technologies and Tools for Full stack development 	R3- Chapter 1, https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/How_the_Web_works





3	Module 3: Web Protocols	Module 3: Web Protocols <ul style="list-style-type: none"> HTTP <ul style="list-style-type: none"> HTTP Request- Response and its structure HTTP Methods- GET, PUT, POST, DELETE HTTP Headers Connection management - HTTP/1.1 and HTTP/2 	T1- Chapter 6 R3- Chapter 5 https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview
4	Module 3: Web Protocols(continued)	Module 3: Web Protocols(continued) <ul style="list-style-type: none"> Synchronous and asynchronous communication Communication with Backend <ul style="list-style-type: none"> AJAX, Fetch API Webhooks Server-Sent Events Polling Bidirectional communication - Web sockets 	T1- Chapter 8
5	Module 4: Server Side: Implementing Web Services	Module 4: Server Side: Implementing Web Services <ul style="list-style-type: none"> REST <ul style="list-style-type: none"> Principles of REST REST constraints Service Design with REST Interaction Design with HTTP Interface Design (URI) Representation and Metadata design 	T2, R4- Chapter 4 R2
6	Module 4: Server Side: Implementing Web Services	Module 4: Server Side: Implementing Web Services <ul style="list-style-type: none"> REST (continued) <ul style="list-style-type: none"> Implementing REST API(NodeJS) <ul style="list-style-type: none"> Using a Framework URL Mapping Routing Requests-Redirection 	R4- Chapter 6 T1- Chapter 13,14, T2





		<ul style="list-style-type: none"> ○ Implementing a web server ○ Processing request, response, data ▪ Storing data in databases(MongoDB/) <ul style="list-style-type: none"> ○ Models ○ Object Relational Mapper ○ Interaction with DBs 	
7	Module 4: Server Side: Implementing Web Services	Module 4: Server Side: Implementing Web Services <ul style="list-style-type: none"> • REST <ul style="list-style-type: none"> ▪ API versioning and documentation ▪ Open API ▪ Using swagger • GraphQL <ul style="list-style-type: none"> ▪ Schemas and Types ▪ Thinking in Graphs ▪ Serving over HTTP ▪ Implementing the GraphQL API ▪ Validation and Execution 	R2, T2, R4- Chapter 5, 8
8	Module 4: Server Side: Implementing Web Services	Module 4: Server Side: Implementing Web Services <ul style="list-style-type: none"> • gRPC <ul style="list-style-type: none"> ▪ Service Definition- Protobuf ▪ Architecture ▪ Channels Streaming and Types 	R5-Chapter 1, 2,3
9	Module 5: Securing Application	Module 5: Securing Application <ul style="list-style-type: none"> • Basic Authentication • API Authorization • JSON Web Tokens • OAuth • SSL, TLS and HTTPS • Common Vulnerabilities 	T1- Chapter 18 R4- Chapter 11
10	Module 6:	Module 6: Understanding Frontend Development	https://developer.mozilla.org/en





	Understanding Frontend Development	<ul style="list-style-type: none"> • Designing and Structuring Webpages • Making pages Interactive with JavaScript • Using Browser based Web APIs- DOM, Web storage • Client-side JavaScript Frameworks: Features and Advantage, MEAN, MERN 	- US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks
11	Module 6: Understanding Frontend Development (continued)	Module 6: Understanding Frontend Development(continued) <ul style="list-style-type: none"> • Implementing Single Page Applications using JavaScript Tech stack(React) <ul style="list-style-type: none"> ▪ The Node Ecosystem ▪ Project Setup ▪ Package managers- NPM, Module bundlers webpack, Build tools ▪ Creating and styling components ▪ Managing Component hierarchies and lifecycle 	R1, https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/React_getting_started
12	Module 6: Understanding Frontend Development (continued)	Module 6: Understanding Frontend Development(continued) <ul style="list-style-type: none"> ▪ Managing State ▪ Routing ▪ Building, deploying and Hosting 	R1, https://react.dev/learn
13	Module 7: Testing	Module 7: Testing <ul style="list-style-type: none"> • API Testing <ul style="list-style-type: none"> ▪ API Testing and Types ▪ Unit Testing ▪ Contract Testing • Frontend <ul style="list-style-type: none"> ▪ Unit testing ▪ Cross browser testing • Acceptance Testing 	R4- Chapter 12
14	Module 8:	Module 8: Accessibility and Performance	R3-Chapter 8, 21





	Accessibility and Performance	<ul style="list-style-type: none"> • Accessibility <ul style="list-style-type: none"> ▪ Inclusive Design ▪ Assistive Technologies ▪ Web content accessibility Guidelines ▪ Optimizing Websites for Accessibility ▪ Testing Accessibility • Performance <ul style="list-style-type: none"> ▪ Tools and Metrics for Measuring Performance ▪ Options for Optimization <ul style="list-style-type: none"> ○ Caching- Client side, Server side ○ Minifying Code, Compressing files ○ Lazy Loading 	
15	Module 9: Latest Advancements	Module 9: Latest Advancements <ul style="list-style-type: none"> • Progressive Web Apps • Web Assembly • Microfrontends 	Classroom discussions
16		<ul style="list-style-type: none"> • Recap 	

Experiential Learning Components:

Objective of Experiential Learning Component:

Lab Sheets: Perform various list of experiments(attached below) to understand the various components of the web application

Work Integrated Learning Assignment: Develop and test a working model of web application using the chosen tech stack for the selected stories.

Scope of Experiential Learning Component:

Implement a functional end-to-end web application using client-side and server-side web technologies.





Lab Infrastructure:

- Visual Studio Code or any preferred editor
- Browser(Chrome)
- Frontend: ReactJS, Node Ecosystem
- Backend: NodeJS, Express, Django
- Database: MongoDB/Postgresql
- Postman for Testing
- GitHub and GIT for source code repository

List of Experiments:

Exp No.	Experiment Title	Reference to handout module/section
1.	Using Browser's Developer Tools to understand flow of HTTP communication	Module 2
2.	Demonstration of HTTP request response and Demonstration of API calls from frontend	Module 3
3.	Demonstration of Webhooks and Demonstration of Web sockets	Module 3
4.	Create REST service to perform CRUD operations with the database, using JSON data format. Create Documentation for the REST APIs	Module 4
5.	Implementaion of a GraphQL API	Module 4
6	Add Authentication with JWT to REST APIs	Module 5
7	Create React Components and Build React app	Module 6
8	Perform unit testing for REST API and frontend React components	Module 7





Evaluation Scheme:

Legend: EC = Evaluation Component; AN = After Noon Session; FN = Fore Noon Session

Evaluation Component	Name (Quiz, Lab, Project, Mid-term exam, End semester exam, etc.)	Type (Open book, Closed book, Online, etc.)	Weight	Duration	Day, Date, Session, Time
EC – 1*	Quiz	Online	10%	1 week	To be updated
	Assignment	Online	20 %	10 days	To be updated
EC - 2	Mid-Semester Test	Closed Book	30%	2 hours	To be updated
EC - 3	Comprehensive Exam	Open Book	40%	2 ½ Hours	To be updated

Syllabus for Mid-Semester Test (Closed Book): Topics in Contact session: 1 to 8

Syllabus for Comprehensive Exam (Open Book): All topics

Important Links and Information:

eLearn Portal: <https://elearn.bits-pilani.ac.in>

Students must visit the eLearn portal regularly and stay updated with the latest announcements and deadlines.

Contact Sessions: Students should attend the online lectures as per the schedule provided on the eLearn portal.

Evaluation Guidelines:

1. EC-1 consists of either two Assignments or three Quizzes. Students will attempt them through the course pages on the eLearn portal. Announcements will be made on the portal in a timely manner.
2. For Closed Book tests: No books or reference material of any kind will be permitted.
3. For Open Book exams: “open book” means text/ reference books (publisher copy only) and does not include any other learning material. No other learning material will be permitted during the open book examinations. For Detailed Guidelines refer to the attached document. [EC3 Guidelines](#)
4. If a student is unable to appear for the Regular Test/Exam due to genuine exigencies, the student should follow the procedure to apply for the Make-Up Test/Exam, which will be made available on the eLearn portal. The Make-Up Test/Exam will be conducted only at selected exam centres on the dates to be announced later.

It shall be the responsibility of the individual student to be regular in maintaining the self-study schedule as given in the course handout, attend the online lectures, and take all the prescribed evaluation components such as Assignments/Quizzes, Mid-Semester Tests and Comprehensive Exams according to the evaluation scheme provided in the handout.

