**BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI**



**WORK INTEGRATED LEARNING PROGRAMMES**

Part A: Content Design

|  |  |
| --- | --- |
| **Course Title** | **Full Stack Application Development** |
| **Course No(s)** | SE ZG503 |
| **Credit Units** | 4 |
| **Content Authors** | Akshaya Ganesan |
| **Version** | 1.0 |
| **Date** | September 2023 |

**Course Description:**

The modern software application landscape is evolving rapidly, moving from the conventional, layered web applications hosted on remote servers to the mobile-only / mobile-first and cloud-native applications with complex deployment options. At the same time, the software development teams have started adapting the most agile development methodologies, enabling them to deliver the software with better quality at shorter and more frequent intervals. These developments have resulted in the necessity of engineers having multiple skill sets essential for every aspect of the software development life cycle, right from requirement analysis to application deployment.

This course aims to provide a comprehensive introduction to modern Web application architecture approaches, frontend and backend technologies, and suitable web application frameworks required for developing modern web apps. It focuses on designing and implementing end-to-end functional web applications, learning the key patterns followed at each layer of the application architecture and technology considerations to choose an appropriate implementation technology. The course also involves hands-on exposure to full-stack development of web applications using development frameworks.

**Course Objectives**

|  |  |
| --- | --- |
| **No** | Course Objective |
| **CO1** | Understand the modern application landscape and the evolution of the application landscape. |
| **CO2** | Build an understanding of an end-to-end application's typical structure, design, and implementation considerations. |
| **CO3** | Comprehend the necessity and usefulness of the client and server-side frameworks, along with their strengths and weaknesses. |
| **CO4** | 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 |

**Reference Book(s) & other resources**

|  |  |
| --- | --- |
| R1 | Building Microservices: Designing Fine-Grained Systems Book by Sam Newman, 1st edition, published by O'Reilly Media, Feb 2015 |
| 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
* Microfronends

**Learning Outcomes:**

|  |  |
| --- | --- |
| No | Learning Outcomes |
| 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**

|  |  |
| --- | --- |
| **Academic Term** | First Semester 2024-2025 |
| **Course Title** | **Full Stack Application Development** |
| **Course No** | SE ZG503 |
| **Lead Instructor** | AKSHAYA GANESAN |

|  |  |  |
| --- | --- | --- |
| Session No. | Topic Title | Study / HW Resource Reference |
| 1 | **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 | R3- Chapter 12,  R1  Web references |
| 2 | **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 | 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**   * HTTP   + 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)**   * Synchronous and asynchronous communication * Communication with Backend   + AJAX, Fetch API   + Webhooks   + Server-Sent Events * Polling * Bidirectional communication - Web sockets | T1- Chapter 8 |
| 5 | **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 | R4- Chapter 4  R2 |
| 6 | **Module 4: Server Side: Implementing Web Services**   * REST (continued) * Implementing REST API(NodeJS/Python)   + Using a Framework   + URL Mapping   + Routing Requests-Redirection   + Implementing a web server   + Processing request, response, data * Storing data in databases(MongoDB/Postgres)   + Models   + Object Relational Mapper   + Interaction with DBs | R4- Chapter 6  T1- Chapter 13,14 |
| 7 | **Module 4: Server Side: Implementing Web Services**   * REST * API versioning and documentation * Open API * Using swagger * GraphQL   + Schemas and Types   + Thinking in Graphs   + Serving over HTTP   + Implementing the GraphQL API   + Validation and Execution | R2,  R4- Chapter 5, 8 |
| 8 | **Module 4: Server Side: Implementing Web Services**   * gRPC   + Service Definition- Protobuf   + Architecture   + Channels Streaming and Types | R5-Chapter 1, 2,3 |
| 9 | **Module 5: Securing Application**   * Basic Authentication * API Authorization * JSON Web Tokens * OAuth * SSL, TLS and HTTPS * Common Vulnerabilities | T1- Chapter 18  R4- Chapter 11 |
| 10 | **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 | https://developer.mozilla.org/en-US/docs/Learn/Tools\_and\_testing/Client-side\_JavaScript\_frameworks |
| 11 | **Module 6: Understanding Frontend Development(continued)**   * Implementing Single Page Applications using JavaScript Tech stack(React/Angular)   + The Node Ecosystem   + Project Setup   + Package managers- NPM, Module bundlers webpack, Build tools   + Creating and styling components   + Managing Component hierarchies and lifecycle | 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)**   * Managing State * Routing * Building, deploying and Hosting | https://react.dev/learn |
| 13 | **Module 7: Testing**   * API Testing   + API Testing and Types   + Unit Testing   + Contract Testing * Frontend   + Unit testing   + Cross browser testing   + Acceptance Testing | R4- Chapter 12 |
| 14 | **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 | R3-Chapter 8, 21 |
| 15 | **Module 9: Latest Advancements**   * Progressive Web Apps * Web Assembly * Microfronends | Classroom discussions |
| 16 | Recap |  |

**Evaluation Scheme**:

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Name | Type | Duration | Weight | Day, Date, Session, Time |
| EC-1 | Quizzes | Online |  | 10% | September 1-10, 2024 |
| EC-1 | Assignments | Take Home |  | 20% | October 10-20, 2024 |
| EC-2 | Mid-Semester Test | Closed Book | 2 Hours | 30% | Sunday, 22/09/2024 (FN) |
| EC-3 | Comprehensive Exam | Open Book | 2.5 Hrs | 40% | Sunday, 01/12/2024 (FN) |

**Note:**

Syllabus for Mid-Semester Test (Closed Book): Topics in Session Nos. 1 to 8

Syllabus for Comprehensive Exam (Open Book): All topics (Session Nos. 1 to 16)

Important links and information:

Elearn portal: https://elearn.bits-pilani.ac.in

Students are expected to visit the Elearn portal on a regular basis and stay up to date 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: Use of books and any printed / written reference material (filed or bound) is permitted. However, loose sheets of paper will not be allowed. Use of calculators is permitted in all exams. Laptops/Mobiles of any kind are not allowed. Exchange of any material is not allowed.
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 Assignment/Quiz, Mid-Semester Test and Comprehensive Exam according to the evaluation scheme provided in the handout.

Evaluation Guidelines:

1. EC-1 consists of two 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. EC-2 consists of either one or two Assignments. Students will attempt them through the course pages on the Elearn portal. Announcements will be made on the portal, in a timely manner.
3. For Closed Book tests: No books or reference material of any kind will be permitted.
4. For Open Book exams: Use of books and any printed / written reference material (filed or bound) is permitted. However, loose sheets of paper will not be allowed. Use of calculators is permitted in all exams. Laptops/Mobiles of any kind are not allowed. Exchange of any material is not allowed.
5. 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 hand-out, attend the online lectures, and take all the prescribed evaluation components such as Assignment/Quiz, Mid-Semester Test and Comprehensive Exam according to the evaluation scheme provided in the hand-out.