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

**WORK INTEGRATED LEARNING PROGRAMMES**

**Digital Learning**

**Part A: Course Design**

|  |  |
| --- | --- |
| **Course Title** | Cross Platform Application Development |
| **Course No(s)** | SE ZG585/SS ZG585 |
| **Credit Units** | 4 |
| **Content Authors** | Pravin Y Pawar |
| **Version** | 2.0 |
| **Revision Date** | 01 February 2024 |

**Course Description**

|  |
| --- |
| Cross-platform applications development involves creation of software applications that are compatible with multiple platforms or software environments. This course aims to equip students with the expertise to design and develop web and mobile based applications that can operate in varied environments and platforms. Additionally, it also aims to develop the understanding of the role and importance of API management in such applications. The course involves hands-on exposure to full stack development of cross-platform applications using some of the existing development frameworks. |

**Course Objectives**

|  |  |
| --- | --- |
| **The course aims at:** | |
| **CO1** | Introducing the modern application landscape ranging from web, mobile apps to cloud native, Serverless apps |
| **CO2** | Exploring the frameworks, tools choices available for various types of cross platform applications such as native, hybrid , web and multiplatform apps |
| **CO3** | Developing multiplatform application with leading edge application framework |
| **CO4** | Identifying the need, architectural styles, design considerations and management essentials for the Application Programming interfaces (APIs) |

**Text Book(s)**

|  |  |
| --- | --- |
| **T1** | Continuous API Management – Making the right decisions in evolving landscape, Medjaoui, Wilde, Mitra, Amundsen, O’REILY |

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

|  |  |
| --- | --- |
| **R1** | Designing Web APIs – Building APIs that developers love, Jin, Sahni, Shevat , O’REILY |
| **R2** | Various product, tools, frameworks documentation |

**Learning Outcomes:**

|  |  |
| --- | --- |
| **Students will :** | |
| **LO1** | Get an overview of modern application paradigms, respective architectures and development framework options |
| **LO2** | Appreciate the necessity and usage of modern edge cloud based application platforms required for Serverless apps |
| **LO3** | Obtain hands-on experience in multiplatform application design and development using the cutting edge framework involving user interface, interaction with server side etc. |
| **LO4** | Realize the need for API management, challenges involved therein and considerations for the same |

**Part B: Course Handout**

|  |  |
| --- | --- |
| **Academic Term** | First Semester 2024-2025 |
| **Course Title** | Cross Platform Application Development |
| **Course No** | SE ZG585/SS ZG585 |
| **Lead Instructor** | CHANDAN RN |

**Glossary of Terms**

|  |  |  |
| --- | --- | --- |
| **Module** | **M** | Module is a standalone quantum of designed content. A typical course is delivered using a string of modules. M2 means module 2. |
| **Contact Hour** | **CH** | Contact Hour (CH) stands for an hour long live session with students conducted either in a physical classroom or enabled through technology. In this model of instruction, instructor led sessions will be for 32 CH. |
| **Recorded Lecture** | **RL** | RL stands for Recorded Lecture or Recorded Lesson. It is presented to the student through an online portal. A given RL unfolds as a sequences of video segments interleaved with exercises. |
| **Lab Exercises** | **LE** | Lab exercises associated with various modules |
| **Self-Study** | **SS** | Specific content assigned for self-study |
| **Homework** | **HW** | Specific problems/design/lab exercises assigned as homework |

**Modular Structure**

**Module Summary**

|  |  |
| --- | --- |
| **No.** | **Content of the Module** |
| M1 | Modern Application Landscape   * Web apps * Mobile applications * Cross Platform applications * Cloud native applications * Serverless Apps * API Products |
| M2 | Modern Architectural Styles   * Modern app requirements * Architectural styles overview |
| M3 | Cross Platform Mobile Applications Development   * Native Applications   + Platforms – Android, iOS etc.   + Framework choices, benefits, limitations * Cross Platform – Native Applications   + Motivation, working   + Framework Choices (ReactNative, Xamarin, Flutter etc.) * Cross Platform - Web Apps   + Purpose, working   + Framework Choices (Ionic, Cordova/PhoneGap, Capacitor) |
| M4 | Serverless Apps   * BaaS/mBaaS   + Motivation, Choices (Firebase / Parse / Back4App ) * FaaS   + Motivation, Options (AWS Lambda / Google Cloud Functions / Azure Functions ) |
| M5 | Low Code Applications   * Motivation, use cases, choices * AI-Powered Low-Code Tools |
| M6 | API Management   * API as a Product   + Challenge of API Management   + API as a Product lifecycle   + Continuous API improvement * API Landscape   + API teams   + API Management at scale   + Managing API lifecycle in an Evolving Landscape |

**Detailed Structure**

**M1: Modern Application Landscape**

**Contact Session 1-2**

|  |  |  |  |
| --- | --- | --- | --- |
| Session | Type | Description/Plan | Reference |
| 1-2 | CH1-2 | **Modern Applications**   * Web apps   + Conventional "N-tier" applications   + Single page applications (SPA) * Mobile applications | * Classroom Discussions |
| CH3-4 | * Cross Platform applications * Cloud native applications * Serverless Apps * API Products | * Classroom Discussions |
| Post CS | SS1 | * Explore more on the   + Cloud native applications needs   + Serverless applications types   + Types of APIs |  |

**M2: Modern Architectural Styles**

**Contact Session 3**

|  |  |  |  |
| --- | --- | --- | --- |
| Session | Type | Description/Plan | Reference |
| 3 | CH 5-6 | **Modern Architectures**   * Today’s application requirements * Modern Architectures * Architectural Styles Overview & Constraints | * Classroom Discussions |
| Post CS | SS2 | * Survey the modern age applications and note down the commonalities among them |  |

**M3: Cross Platform Mobile Applications Development**

**Contact Session 4-9**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Session | Type | Description/Plan | | Reference |
| 4-5 | CH7-10 | **Native Applications**   * Platforms – Android, iOS etc. * Framework choices, benefits, limitations   + Android/Java, Android /Kotlin, iOS/Swift * Comparison of Platforms | | * Classroom Discussions * Developer documentation |
| 6-7  8-9 | CH11-14 | **Cross Platform – Native Apps**   * Motivation, working * Framework Choices   + ReactNative, Xamarin, NativeScript, Flutter etc. * Comparison of frameworks | | * Classroom Discussions * Developer documentation |
| CH15-18 | **Cross Platform - Web Apps**   * Purpose, working * Framework Choices   + Ionic, Cordova/PhoneGap, Capacitor * Comparison of framework | | * Classroom Discussions * Developer documentation |
| Post CS | SS3.1 | * List down the application that are most suitable candidate for native app development * Explore more about AWS device farm which allows testing of mobile apps on cloud platform * Think about what sort of CI/CD pipeline will be required for the mobile app development |  | |
| SS3.2 | * Study about how organizations are getting benefitted through cross platform application platforms development frameworks * Prepare a simple calendar application using one of the framework discussed in the class, leverage the database for data storage |  | |
| SS3.3 | * Explore more about the various ecosystem components provided by the Ionic * List down the scenarios where cross platform web apps outshine the other types of mobile apps |  | |
| LE3 | * Mobile Apps Development with Flutter * Google Maps with Flutter app * Flutter app with persistence | * Lab 1 manual * Lab 2 manual * Lab 3 manual | |

**M4: Serverless Apps**

**Contact Session 10-11**

|  |  |  |  |
| --- | --- | --- | --- |
| Session | Type | Description/Plan | Reference |
| 10 | CH19-20 | **Serverless Apps with BaaS**   * Motivation * Choices   + Firebase / Parse / Back4App * Demonstration | * Classroom Discussions |
| 11 | CH21-22 | **Serverless Apps with FaaS**   * Motivation * Options   + AWS Lambda / Google Cloud Functions / Azure Functions * Demonstration | * AWS Docs * GCP Docs * Azure Docs |
| Post CS | SS4.1 | * Explore more about Firebase and how it helps in speeding up the mobile apps development |  |
| SS4.2 | * Try out the getting started guides provided by different cloud service providers for the FaaS services |  |
| LE4 | * Serverless App Development | * Lab 4 manual |

**M5: Low Code Applications**

**Contact Session 12**

|  |  |  |  |
| --- | --- | --- | --- |
| Session | Type | Description/Plan | Reference |
| 12 | CH 23-24 | **Low code Architectures**   * Motivation * Platform choices * Demonstration | * Classroom Discussions |
| 13 | CH 25-26 | **AI-Powered Low-Code Tools**   * Microsoft Power Platform | * Classroom demonstrations |
| Post CS | SS5.1 | * Explore how Serverless apps are different than No/Low code apps |  |

**M6: API Management**

**Contact Session 14-15**

|  |  |  |  |
| --- | --- | --- | --- |
| Session | Type | Description/Plan | Reference |
| 14 | CH27 | **API Product Management**   * Challenge of API Management * API as a Product lifecycle * Pillars of API product * Continuous API improvement | * T1 – Ch1,3, 5,6 |
|  | CH28 |
| 15 | CH29 | **API Landscape**   * API teams * API Management at scale * API platforms * API Analytics | * T1 – Ch 7,8,10 |
| CH30 |
| Post CS | SS6.1  LE6 | * Compare the APIM capabilities offered by the different providers * API Management with Microsoft Azure | * Lab 5,6, 7, 8 manual |

**Contact Session 16**

|  |  |  |  |
| --- | --- | --- | --- |
| Session | Type | Description/Plan | Reference |
| 16 | CH31-32 | * Review |  |

**Experiential Leaning Component**

|  |  |  |  |
| --- | --- | --- | --- |
| **Lab** | **Topic** | **Description** |  |
| 1 | Mobile Apps Development with Flutter | * Write a Flutter app that looks natural on iOS, Android, and the web * Building layouts | * Virtual Labs |
| 2 | Google Maps with Flutter app | * Display a Google map in an app * Display the data as markers on the map. | * Virtual Labs |
| 3 | Flutter app with persistence | * Implement a simple app interacting with local database | * Virtual Labs |
| 4 | Serverless App Development | * Serverless App development using AWS managed services like Lambda, Fargate etc. | * Virtual Labs |
| 5 | API Management with Microsoft Azure – I  Azure API Management Service | * Getting Started with API Management with Microsoft Azure | * Virtual Labs |
| 6 | API Management with Microsoft Azure – II  Managing API Products | * Learn to create an API product * Study to use APIM to create a blank API and manage it manually, then set a policy on an API so it returns a mocked response. | * Virtual Labs |
| 7 | API Management with Microsoft Azure – III  Monitoring an API | * Learn to transform your API so it does not reveal info about the private backend * Describe how easy it is to add protection for your backend API by configuring a rate limit with Azure API Management * Shows how to visualize, query, route, archive, and take actions on the metrics or logs coming from your Azure API Management service | * Virtual Labs |
| 8 | API Management with Microsoft Azure – IV  Managing Versions of API | * Describes steps to make revisions to existing APIs and deploy them for developers usage * Demonstrates how easy it is to add version of API | * Virtual Labs |

**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** | Experiential learning Assignment-I | Take Home | 15 days | 10% | September 1-10, 2024 |
| Experiential learning Assignment-II | Take Home | 15 days | 20% | October 10-20, 2024 |
| **EC-2** | Mid-Semester Test | Closed Book | 2 hours | 30% | Friday, 20/09/2024 (FN) |
| **EC-3** | Comprehensive Exam | Open Book | 2 ½ hours | 40% | Friday, 29/11/2024 (FN) |

Syllabus for Mid-Semester Test (Closed Book): Topics in Session Nos. 1 to 7

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. EC1 consists of two assignments. Announcements will be made available 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.