



BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
WORK INTEGRATED LEARNING PROGRAMMES

Digital

Part A: Content Design

| | |
|------------------------|---|
| Course Title | CROSS PLATFORM APPLICATION DEVELOPMENT |
| Course No(s) | SE* ZG585 |
| Credit Units | 4 |
| Credit Model | |
| Content Authors | Chandan R N |
| Version | 2.0 (Updated as per new instructions model) |

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 | Developing understanding about the typical structure, design and implementation considerations of an end-to-end application |
| CO3 | Exploring the frameworks, tools choices available for various types of cross platform applications such as native, hybrid , web and multiplatform apps |
| CO4 | Developing multiplatform application with leading edge application framework |
| CO5 | Identifying the need, architectural styles, design considerations and management essentials for the Application Programming interfaces (APIs) |

Text Book(s)

| | |
|-----------|---|
| T1 | Designing Web APIs – Building APIs that developers love, Jin, Sahni, Shevat , O'REILY |
| T2 | Continuous API Management – Making the right decisions in evolving landscape, |



| | |
|--|---|
| | Medjaoui, Wilde, Mitra, Amundsen, O'REILY |
|--|---|

Reference Book(s) & other resources

| | |
|-----------|---|
| R1 | Building Microservices – designing fine grained systems , O'REILY |
| R2 | API Architecture, Matthias Biehl |
| R3 | 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 microservices and 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 | Recognize the role, usage of APIs in applications and experience the design and development of APIs using the commonly used ecosystem tools |
| LO5 | Realize the need for API management, challenges involved therein and considerations for the same |

Part B: Learning Plan

| | |
|------------------------|--|
| Academic Term | SECOND SEM 2022-2023 |
| Course Title | CROSS PLATFORM APPLICATION DEVELOPMENT |
| Course No | (Merged - SEZG585/SSZG585)(S1-22) |
| Lead Instructor | Chandan R N |

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 a 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 | RL | RL stands for Recorded Lecture or Recorded Lesson. It is |



| | | |
|----------------------|-----------|--|
| Lecture | | 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 <ul style="list-style-type: none"> • Web apps • Mobile applications • Cross Platform applications • Cloud native applications • Serverless Apps • API Products |
| M2 | App Structure <ul style="list-style-type: none"> • Typical structure of end-to-end application <ul style="list-style-type: none"> ○ Frontend and Backend components, Databases ○ Interaction between the components • Modern Architectural Styles <ul style="list-style-type: none"> ○ Modern app requirements ○ Architectural styles overview |
| M3 | Modern App Architectures <ul style="list-style-type: none"> • Microservices <ul style="list-style-type: none"> ○ Motivation, Architecture, Constraints ○ Migrating legacy applications • Cloud Native Architecture [Optional]# <ul style="list-style-type: none"> ○ Distributed Systems and Fallacies, 12 factor app ○ Cloud based architecture ○ Motivation, Principles, Architecture, Components, Constraints • Serverless Architecture <ul style="list-style-type: none"> ○ Deployment approaches ○ Cloud services for development, testing and deployments of applications (P/B/FaaS) • Low Code Architecture [Optional] # <ul style="list-style-type: none"> ○ Motivation, use cases, choices |
| M4 | Serverless Apps <ul style="list-style-type: none"> • BaaS/mBaaS <ul style="list-style-type: none"> ○ Motivation, Choices (Firebase / Parse / Back4App) • FaaS <ul style="list-style-type: none"> ○ Motivation, Options (AWS Lambda / Google Cloud Functions / Azure Functions) |



| | |
|----|---|
| M5 | Cross Platform Mobile Applications Development <ul style="list-style-type: none">• Native Applications<ul style="list-style-type: none">○ Platforms – Android, iOS etc.○ Framework choices, benefits, limitations• Cross Platform – Native Applications<ul style="list-style-type: none">○ Motivation, working○ Framework Choices (ReactNative, Xamarin, Flutter etc.)• Cross Platform - Web Apps<ul style="list-style-type: none">○ Purpose, working○ Framework Choices (Ionic, Cordova/PhoneGap, Capacitor) |
| M6 | API Design <ul style="list-style-type: none">• Application Programming Interfaces<ul style="list-style-type: none">○ Motivations, requirements, constraints○ API Paradigms○ API Platform Architecture• API Design Practices<ul style="list-style-type: none">○ Design of API○ Best practices○ Developer resources• API Specification / Description<ul style="list-style-type: none">○ Usage○ Languages |
| M7 | API Management <ul style="list-style-type: none">• API as a Product<ul style="list-style-type: none">○ Challenge of API Management○ API as a Product lifecycle○ Continuous API improvement• API Landscape<ul style="list-style-type: none">○ API teams○ API Management at scale○ Managing API lifecycle in an Evolving Landscape |

Refer the courseware page for the optional topics / videos

Detailed Lecture Plan

M1: Modern Application Landscape

| Type | Description/Plan/Reference |
|----------|---|
| RL_1.1.* | <ul style="list-style-type: none">• Web apps |
| RL_1.2.* | <ul style="list-style-type: none">• Mobile applications |
| RL_1.3.* | <ul style="list-style-type: none">• Cross Platform applications |
| RL_1.4.* | <ul style="list-style-type: none">• Cloud native applications |
| RL_1.5.* | <ul style="list-style-type: none">• Serverless Apps |
| RL_1.6.* | <ul style="list-style-type: none">• API Products |



| | | |
|--------|---|--|
| CS 1 | <ul style="list-style-type: none"> • Single Page Applications, MVC • Mobile Apps - Landscape • Cross Platform Apps - Comparison with Native Apps | <ul style="list-style-type: none"> • Classroom discussion |
| CS 2 | <ul style="list-style-type: none"> • Cross Platform Apps - Selection of app type • Cloud native Landscape • Serverless Computing • API Products | <ul style="list-style-type: none"> • Classroom discussion |
| SS 1.1 | <ul style="list-style-type: none"> • Identify the factors that needs to be taken into consideration while developing mobile apps • Note down the common APIs those are used in applications that we daily use • Explore more on the <ul style="list-style-type: none"> ✓ Microservices ✓ Serverless applications types ✓ Types of APIs | |

M2: App Structure

| Type | Description/Plan/Reference | |
|----------|--|--|
| RL_2.1.* | <ul style="list-style-type: none"> • Frontend components | |
| RL_2.2.* | <ul style="list-style-type: none"> • Backend components | |
| RL_2.3.* | <ul style="list-style-type: none"> • Databases | |
| RL_2.4.* | <ul style="list-style-type: none"> • Modern Architectural Styles | |
| CS 3 | <ul style="list-style-type: none"> • Typical structure of end-to-end application <ul style="list-style-type: none"> ○ Frontend and Backend components, Databases ○ Interaction between the components | <ul style="list-style-type: none"> • Classroom discussion |
| CS 4 | <ul style="list-style-type: none"> • Modern Architectural Styles <ul style="list-style-type: none"> ○ Modern app requirements ○ Architectural styles overview | <ul style="list-style-type: none"> • Classroom discussion |
| SS 2.1 | <ul style="list-style-type: none"> • Explore more on the differences between the Relational and Non-relational databases • Identify the options available for the caching systems • Survey the modern age applications and note down the commonalities among them | |



M3: Modern App Architectures

| Type | Description/Plan/Reference | |
|-----------------------------------|--|--|
| 3.1 Modern App Architectures - I | | |
| RL_3.1.* | <ul style="list-style-type: none">Microservices Architecture | |
| RL_3.2.* | <ul style="list-style-type: none">Cloud Native Architecture [Optional] | |
| CS 5 | <ul style="list-style-type: none">Microservices<ul style="list-style-type: none">Motivation, Architecture, ConstraintsMigrating legacy applications | <ul style="list-style-type: none">Classroom discussion |
| CS 5 | <ul style="list-style-type: none">Cloud Native Architecture [Optional]<ul style="list-style-type: none">Distributed Systems and Fallacies, 12 factor appCloud based architectureMotivation, Principles, Architecture, Components, Constraints | <ul style="list-style-type: none">Classroom discussion |
| SS 3.1 | <ul style="list-style-type: none">Explore and note down the challenges associated with microservices based architectureStudy that how cloud application model is more suitable for distributed application development | |
| 3.2 Modern App Architectures - II | | |
| RL_3.3.* | <ul style="list-style-type: none">Serverless Architecture | |
| RL_3.4.* | <ul style="list-style-type: none">Low Code Architecture [Optional] | |
| CS 6 | <ul style="list-style-type: none">Serverless Architecture<ul style="list-style-type: none">Deployment approachesCloud services for development, testing and deployments of applications (P/B/FaSS) | <ul style="list-style-type: none">Classroom discussion |
| CS 6 | <ul style="list-style-type: none">Low Code Architecture [Optional]<ul style="list-style-type: none">Motivation, use cases, choices | <ul style="list-style-type: none">Classroom discussion |
| SS 3.2 | <ul style="list-style-type: none">Discuss the advantages and disadvantages related to the cloud computing in generalExplore how Serverless apps are different than SaaS appsIdentify what are the common use cases in our daily life that can be easily targeted through the low code architecture | |



M4: Serverless Apps

| Type | Description/Plan/Reference | |
|----------|---|--|
| RL_4.1.* | <ul style="list-style-type: none"> Backend as a Service (BaaS) | |
| RL_4.2.* | <ul style="list-style-type: none"> Function as a Service (FaaS) [Optional] | |
| CS 7 | <ul style="list-style-type: none"> BaaS/mBaaS <ul style="list-style-type: none"> Motivation, Choices (Firebase / Parse / Back4App) Demonstration | <ul style="list-style-type: none"> Classroom discussion |
| CS 8 | <ul style="list-style-type: none"> FaaS <ul style="list-style-type: none"> Motivation, Options (AWS Lambda / Google Cloud Functions / Azure Functions) Demonstration | <ul style="list-style-type: none"> Classroom discussion |
| SS 4.1 | <ul style="list-style-type: none"> Explore more about Firebase and how it helps in speeding up the mobile apps development Try out the getting started guides provided by different cloud service providers for the FaaS services | |
| LE 1 | <ul style="list-style-type: none"> Serverless App Development | <ul style="list-style-type: none"> Lab 1 manual |

M5: Cross Platform Mobile Applications Development

| Type | Description/Plan/Reference | |
|----------------------------------|--|--|
| 5.1 Native Applications | | |
| RL_5.1.* | <ul style="list-style-type: none">• Mobile Apps Development | |
| RL_5.2.* | <ul style="list-style-type: none">• Native Applications | |
| CS 9 | <ul style="list-style-type: none">• Native Applications<ul style="list-style-type: none">○ Platforms – Android, iOS etc.○ Framework choices, benefits, limitations | <ul style="list-style-type: none">• Classroom discussion |
| Tutorial 1 | <ul style="list-style-type: none">○ Demonstration of Android app development using Dart | <ul style="list-style-type: none">• Google Docs |
| SS 6.1 | <ul style="list-style-type: none">• 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 | |
| 5.2 Cross Platform - Native Apps | | |
| RL_5.3.* | <ul style="list-style-type: none">• Cross Platform-Native Apps | |



| | | |
|--------------------------------------|---|--|
| CS 10 | <ul style="list-style-type: none"> • Cross Platform – Native Applications <ul style="list-style-type: none"> ○ Motivation, working ○ Framework Choices (ReactNative, Xamarin, Flutter etc.) | <ul style="list-style-type: none"> • Classroom discussion |
| Tutorial 2 | <ul style="list-style-type: none"> • Demonstration of React native application illustrating the frontend and backend interaction | <ul style="list-style-type: none"> • ReactNative docs |
| SS 7.1 | <ul style="list-style-type: none"> • 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 | |
| LE 2 LE 3 LE 4 | <ul style="list-style-type: none"> • Google Maps with Flutter app • Google Maps with Flutter app • Flutter app with persistence | <ul style="list-style-type: none"> • Lab 2 manual • Lab 3 manual • Lab 4 manual |
| 5.3 Cross Platform – Web Apps | | |
| RL_5.4.* | <ul style="list-style-type: none"> • Cross Platform-Web Apps | |
| CS 11 | <ul style="list-style-type: none"> • Cross Platform - Web Apps <ul style="list-style-type: none"> ○ Purpose, working ○ Framework Choices (Ionic, Cordova/PhoneGap, Capacitor) | <ul style="list-style-type: none"> • Classroom discussion |
| Tutorial 3 | <ul style="list-style-type: none"> • Demonstration of Ionic application illustrating the frontend development using one of commonly used frontend framework | <ul style="list-style-type: none"> • Ionic docs |
| SS 8.1 | <ul style="list-style-type: none"> • 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 | |

M6: API Design

| Type | Description/Plan/Reference | |
|----------|---|--|
| RL_6.1.* | <ul style="list-style-type: none"> • API | |
| RL_6.2.* | <ul style="list-style-type: none"> • API Paradigm | |
| RL_6.3.* | <ul style="list-style-type: none"> • API Platform Architecture | |
| RL_6.4.* | <ul style="list-style-type: none"> • API Design Best Practices | |
| RL_6.5.* | <ul style="list-style-type: none"> • API Description Languages | |
| CS 12 | <ul style="list-style-type: none"> • Application Programming Interfaces <ul style="list-style-type: none"> ○ Motivations, requirements, constraints ○ API Paradigms | <ul style="list-style-type: none"> • T1 Ch 1, 2 |



| | | |
|--------|--|--|
| CS 13 | <ul style="list-style-type: none"> API Specification / Description <ul style="list-style-type: none"> Usage Languages API Design Best Practices | <ul style="list-style-type: none"> R2 – Ch6 |
| SS 9.1 | <ul style="list-style-type: none"> Note down the best practices applied during web API development Explore more about the API description languages Import a simple OpenAPI spec into AWS API gateway and try to create a proxy API which can be invoked from the client side | |
| LE 5 | <ul style="list-style-type: none"> API design and documentation with OpenAPI | <ul style="list-style-type: none"> Lab 5 manual |
| LE 6 | <ul style="list-style-type: none"> GraphQL Getting Started | <ul style="list-style-type: none"> Lab 6 manual |

M7: API Management

| Type | Description/Plan/Reference | |
|------------|---|--|
| RL_7.1.* | <ul style="list-style-type: none"> API Product management | |
| RL_7.2.* | <ul style="list-style-type: none"> API Change Management | |
| RL_7.3.* | <ul style="list-style-type: none"> API teams | |
| RL_7.4.* | <ul style="list-style-type: none"> API management platforms | |
| RL_7.5.* | <ul style="list-style-type: none"> API Analytics | |
| CS 14 | API Product Management <ul style="list-style-type: none"> Challenge of API Management API as a Product lifecycle Pillars of API product Continuous API improvement | <ul style="list-style-type: none"> T2 – Ch1,3,5,6 |
| CS 15 | API Landscape <ul style="list-style-type: none"> API teams API Management at scale API platforms API Analytics | <ul style="list-style-type: none"> T2 – Ch 7,8,10 |
| SS 10.1 | <ul style="list-style-type: none"> Compare the APIM capabilities offered by the different providers | |
| Tutorial 4 | API Management with Microsoft Azure <ul style="list-style-type: none"> Azure API Management Service Managing API Products Monitoring an API Managing Versions of API | <ul style="list-style-type: none"> Lab 7 manual Lab 8 manual Lab 9 manual [Optional] Lab10 manual [Optional] |
| LE 7 | | |
| LE 8 | | |
| LE 9 | | |
| LE 10 | | |

Contact Session 11

| Session | Type | Description/Plan | Reference |
|---------|------|---|-----------|
| 16 | CS | <ul style="list-style-type: none"> Review / Buffer | |

Proposed Structure for Tutorials:

| # | Date/Time | Topic | Quiz (5%) | Remark |
|---|-----------|---------------------------------|-----------|---|
| 1 | TBA | Native Application Development | 15 mins | Best two out of four will be considered |
| 2 | TBA | React Application Development | 15 mins | |
| 3 | TBA | Flutter Application Development | 15 mins | |
| 4 | TBA | Azure API Management | 15 mins | |

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 | Assignment-1 | | - | 25% | February 13-23, 2023 |
| | Quiz-1 | | - | 05% | March 20-30, 2023 |
| EC-2 | Mid-Semester Test | Open Book | 2 hours | 30% | Friday, 10/03/2023 (FN) |
| EC-3 | Comprehensive Exam | Open Book | 2 ½ hours | 40% | Friday, 19/05/2023 (FN) |

Notes:

Syllabus for Mid-Semester Test (Open Book): Topics in M1 to M4 (contact sessions 1 to 8)

Syllabus for Comprehensive Exam (Open Book): All topics including Tutorials

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:

- EC-1 consists of single Assignment and four Quizzes (best two will be considered). Students will attempt them through the course pages on the Elearn portal. Announcements will be made on the portal, in a timely manner.
- For Closed Book tests: No books or reference material of any kind will be permitted.
- 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.