



**Maharashtra Education Society's  
ABASAHEB GARWARE COLLEGE  
(Autonomous)**

*(Affiliated to Savitribai Phule Pune University)*

**Two Years Post-Graduation Programme in Computer Application  
(Faculty of Science & Technology)**

**Syllabus under NEP  
M.Sc. II (Computer Application)**

**To be implemented from Academic Year 2024-2025**

### Structure of the Course: M. Sc. II (Computer Application)

Year	Semester	Course Type	Course Code	Course Title	Remark	Credit	No. of hours to be conducted	
2	III	Core	CA-601-MJ	Full Stack Development - II		4	48	
			CA-602-MJ	Big Data		4	48	
			CA-603-MJ	Software Testing		4	48	
			CA-604-MJP	Lab course based on CA-601-MJ and CA-603-MJ		2	4 per batch per week	
		Elective	CA-610-MJ	Web Services		2	30	
			CA-611-MJP	Lab course based on Web Services		2	4 per batch per week	
			OR					
			CA-612-MJ	DJango		2	30	
			CA-613-MJP	Lab course based on DJango		2	4 per batch per week	
			CA-631-RP	Research Project		4	-	
IV	Core	CA-651-MJP	Industrial Project Design		4	-		
		CA-660-MJ	MOOC		2	-		
	Elective	CA-661-MJ	MOOC		2	-		
			CA-681-OJT	Industrial Training / Industrial Project		14	-	

**SECOND YEAR : SEMESTER I**  
**CA-601-MJ Full Stack Development - II**

**No of Lectures: 48 (Credits-4)**

**Prerequisites:**

- Basics of Java Scripts
- Fundamentals of Full Stack Development
- Concepts of REST and REST APIs
- Fundamentals of databases

**Course Objectives**

1. To introduce React.js
2. To explain component life-cycle and implements hook
3. To understand event handling
4. To introduce routing using React Routers
5. To explain global state management library using Redux
6. To understand Authorization and Authentication
7. To introduce socket programming
8. To explain how to build and deploy React applications

**Learning Outcomes: On completion of this course, students will be able to:**

1. Understand virtual DOM, state, props, functional components and JSX
2. Create custom hook and use hooks as per functionality
3. Apply event handling
4. Create basic routes and dynamic routes
5. Create and use global store along with multiple reducer
6. Apply Authorization and Authentication using JWT token
7. Understand two-way communication between client and server through a web socket
8. Able to deploy React.js applications on server

**Unit 1: Introduction to React.JS**

**08**

- 1.1 React Overview
  - 1.1.1 Basic Structure
  - 1.1.2 Functional Components
  - 1.1.3 Features
  - 1.1.4 Introduction to virtual DOM
  - 1.1.5 Advantages and Limitations
- 1.2 Rendering UI with React
  - 1.2.1 Anatomy of react project
  - 1.2.2 Create a new react application
  - 1.2.3 Templating using JSX
  - 1.2.4 Use of JSX for UI elements
  - 1.2.5 Create react reusable components

<b>Unit 2: Hooks Overview</b>	<b>05</b>
2.1 Concept of Hooks	
2.2 Different types of hooks (useState, useEffect, useReducer, useCallback, useMemo, useRef)	
2.3 Using state and effect hooks	
2.4 Use of React's useState hook to manage state	
2.5 Rules of hooks	
<b>Unit 3: Event Handling in React</b>	<b>02</b>
3.1 Understanding React event system	
3.2 Understanding Synthetic event	
3.3 Working with event handlers	
3.4 Form events	
<b>Unit 4: React Router</b>	<b>05</b>
4.1 Introduction to React Router	
4.2 Understanding Exact Match	
4.3 Route Links	
4.4 useEffect cleanup - handling cancelled request on unmount	
4.5 Dynamic routing - router parameters	
4.6 Fetching data based on router parameters	
<b>Unit 5: State Management Using Redux</b>	<b>10</b>
5.1 Concept of managing the state in applications	
5.2 Concept of middleware and redux cycle	
5.3 Redux principles	
5.4 Create actions, reducer and store it	
5.5 Working on Redux with React Library	
5.6 API store: getState(), dispatch(), and subscribe()	
5.7 Use cases in Redux	
<b>Unit 6: Session Management and JWT</b>	<b>08</b>
6.1 Concept of Authorization and Authentication	
6.2 Types of Authentications and Techniques	
6.3 Concept of JWT	
6.4 JWT Authentication	

## **Unit 7: Redux Toolkit and RTK Query**

**08**

- 7.1 Redux Toolkit
  - 7.1.1 Migration to modern Redux
  - 7.1.2 Installation
  - 7.1.3 Why Redux toolkit over Redux ?
  - 7.1.4 What is slice?
  - 7.1.5 What is configureStore(), createReducer(), create action (), createSlice(), combine slices ()
  - 7.1.6 Advantage of Redux toolkit over Redux
- 7.2 RTK Query
  - 7.2.1 What is RTK query?
  - 7.2.2 Why to use RTK query?
  - 7.2.3 How to use RTK query?
    - CreateApi()
    - FetchBaseQuery()
    - Setup listener()
    - Provider()
    - Use of hooks in component
  - 7.2.4 Advantage and Disadvantage of RTQ query?

## **Unit 8: Deployment and Hosting React Application on Cloud**

**02**

### **References:**

1. Learning React: Functional Web Development with React and Redux, O'REILLY publications
2. Full-Stack React Projects: Learn MERN stack development by building modern web apps using MongoDB, Express, React, and Node.js, 2nd Edition
3. Learning React: A Hands-On Guide to Building Web Applications Using React and Redux, Kirupa Chinnathambi

## CA-602-MJ : Big Data

Lectures: 48 (Credits- 04)

### Prerequisites:

Basic knowledge of Programming Language (preferably Java), Basic knowledge of SQL, exposure to Linux Environment

### Course Objectives:

1. To understand Big Data platform and its Use cases
2. To provide overview of Apache Hadoop
3. To provide understanding of HDFS concepts
4. To Understand concepts of Map Reduce
5. To provide hands on Hadoop EcoSystem

### Learning Outcomes: On completion of this course, students will be able to:

1. Identify Big Data
2. List components of Hadoop and Hadoop Ecosystem
3. Get knowledge of Map Reduce framework
4. To use framework like Pig and Hive to process Big Data

### Unit 1: Introduction to Big Data

10

- 1.1 What is Big Data?
- 1.2 History of Data Management – Evolution of Big Data
- 1.3 Characteristics of Big Data (3V's of Big Data)
- 1.4 Challenges of Big Data
- 1.5 Approaches used traditionally to solve data volume related challenges
- 1.6 NoSQL
  - 1.6.1 Sharding, Replication, Combination of Sharding and Replication
- 1.7 Big Data Analytics
  - 1.7.1 Need for Big Data Analytics
  - 1.7.2 Stages in Big Data Analytics
  - 1.7.3 Types of Big Data Analytics
- 1.8 Domains for Big Data Analytics
  - 1.8.1 Big Data Analytics in Healthcare, Telecom, Finance, Logistics, Retail, Government etc.

### Unit 2: Introduction to Hadoop

10

- 2.1 Introducing Hadoop, RDBMS versus Hadoop
- 2.2 Distributed Computing Challenges, History and overview of Hadoop,
- 2.3 Use Case of Hadoop
- 2.4 Processing Data with Hadoop
- 2.5 Interacting with Hadoop Ecosystem
- 2.6 HDFS (Hadoop Distributed File System)

2.7 Processing Data with Hadoop

<b>Unit 3: Hadoop Distributed File System (HDFS)</b>	<b>08</b>
3.1 The Design of HDFS,	
3.2 HDFS Concepts, Basic Filesystem Operations,	
3.3 Hadoop Filesystems.	
3.4 The Java Interface- Reading Data from a Hadoop URL	
3.5 Reading Data Using the Filesystem API, Writing Data.	
3.6 Data Flow- Anatomy of a File Read, Anatomy of a File Write	
3.7 Managing Resources and Applications with Hadoop YARN (Yet Another Resource Negotiator)	
3.8 Limitations.	
<b>Unit 4: Understanding Map Reduce Fundamentals</b>	<b>08</b>
4.1 The MapReduce Framework	
4.2 Mapper, Reducer, Combiner, Partitioner	
4.3 Searching, Sorting, Compression	
4.4 Techniques to Optimize MapReduce Jobs	
4.5 Uses of MapReduce	
4.6 Role of HBase in Big Data Processing	
4.7 Developing Simple MapReduce Application	
4.8 Points to Consider while Designing MapReduce	
<b>Unit 5: Hive</b>	<b>08</b>
5.1 Introducing Hive	
5.2 Getting Started with Hive	
5.3 Hive Services	
5.4 Data Types in Hive	
5.5 Built-In Functions in Hive	
5.6 Hive DDL	
5.7 Data Manipulation in Hive	
5.8 Data Retrieval Queries	
5.9 Using JOINS in Hive	
<b>Unit 6: Pig</b>	<b>08</b>
6.1 Introducing Pig	
6.2 Running Pig	
6.3 Getting Started with Pig Latin	
6.4 Working with Operators in Pig	
6.5 Debugging Pig	
6.6 Working with Functions in Pig	
6.7 Error Handling in Pig	

## **Unit 7: Big Data Analysis**

**08**

- 7.1 Hbase, data model and implementations,
- 7.2 Hbase clients, Hbase examples,
- 7.3 Praxis, Cassandra,
- 7.4 Cassandra data model, Cassandra examples, Cassandra clients,
- 7.5 Hadoop integration,

### **Reference Books:**

1. Seema Acharya, Subhashini Chellappan, —"Big Data and Analytics", Wiley Publications, 2<sup>nd</sup> Edition, 2014
2. Tom White, —"Hadoop: The Definitive Guide", O'Reilly, 3rd Edition, 2012.
3. DT Editorial Services - "Big Data, Black Book: Covers Hadoop 2, MapReduce, Hive, YARN, Pig, R and Data Visualization"



# CA-603-MJ : Software Testing

Lectures: 48 (Credits- 04)

## Prerequisites:

- Basic understanding in any preferred programming language and web services.

## Course Objectives:

1. To provide the knowledge of fundamental concepts in software testing and skills to design test case plans for testing software.
2. To provide the knowledge of automation testing.
3. To gain the knowledge of selenium IDE
4. To learn the working of different components.
5. To learn how selenium Grid works.
6. To learn how to execute the test cases in selenium using TestNG.
7. To learn the concept of Page Object Model with examples.

## Learning Outcomes: On completion of this course, students will be able to:

1. Understand the fundamental concepts in software testing, design test cases and test plan, review reports of testing for qualitative software.
2. Identify the needs of software test automation and define and develop the test tool to support test automation.
3. Understand the selenium tool and use of different commands.
4. Developers who can immediately set upon rectifying the errors and return the code to the testers.
5. Understand the selenium Grid.
6. Learn how to execute the test cases in selenium and generate reports using TestNG.
7. Understand the concept of Page Object Model.

## Unit 1: Introduction to Testing

10

- 1.1 Introduction to Software Testing
- 1.2 Objectives of Software Testing
- 1.3 STLC- Software Testing Life Cycle.
- 1.4 Types of Testing-Unit Testing
  - 1.4.1 Integration Testing
  - 1.4.2 System Testing
  - 1.4.3 White Box Testing
  - 1.4.4 Black Box Testing
  - 1.4.5 Alpha Testing
  - 1.4.6 Beta Testing
  - 1.4.7 Regression Testing
  - 1.4.8 Performance Testing
  - 1.4.9 User Acceptance Testing (UAT)
- 1.5 Functional testing vs. non-functional testing, Traceability

- 1.6 Test Templates creation and use-Test Plan
  - 1.6.1 Test case template
  - 1.6.2 Design entry and exit criteria for test case
  - 1.6.3 Design test cases in excel.
- 1.7 Test Case design Techniques
  - 1.7.1 Boundary Value Analysis
  - 1.7.2 Equivalence Partitioning

**Unit 2: Introduction to Automation Testing and Selenium** **03**

- 2.1 Concept of Automation testing
- 2.2 Advantages and Disadvantages of Automation testing
- 2.3 History of Selenium
- 2.4 Use of Selenium tool
- 2.5 Differences between Selenium and other Tools like QTP
- 2.6 Different components in Selenium
  - 2.6.1 Selenium IDE
  - 2.6.2 Selenium RC
  - 2.6.3 Selenium Webdriver and Selenium Grid

**Unit 3: Introduction to Selenium IDE** **06**

- 3.1 Installation and Introduction to IDE
- 3.2 Creating first script using record and playback
- 3.3 Installation of Inspector Tools and its Uses
- 3.4 Generating Scripts using different Web Elements
- 3.5 Handling Wait Commands, Validations Commands and Store Commands
- 3.6 Limitations of IDE

**Unit 4: Introduction to Selenium WebDriver** **10**

- 4.1 First Program on selenium
- 4.2 Basic features of Webdriver like get and navigation functions, different types of waits
- 4.3 Verify Page title in Selenium Webdriver
- 4.4 Navigation, Radio button and Checkbox in Selenium Webdriver
- 4.5 Handling Auto Suggestion, Drop-down List, File upload using Sikuli/AutoIT, Drag and Drop in Selenium, Mouse Hover, Keyword Events using Action class, Multiple windows, Alert Messages, Frames, Web Table, Web Calendar
- 4.6 Synchronization using Webdriver (Waits)
- 4.7 Single and multiple screenshots using Selenium
- 4.8 Capture Error message using Webdriver, Cross Browsing using Selenium
- 4.9 Cross Browsing using Selenium
- 4.10 Complete details of Dynamic XPath in Selenium and CSS in Selenium
- 4.11 Download files in Selenium Webdriver
- 4.12 Use of Logs File and Properties File in Selenium

## **Unit 5: Selenium Grid**

**10**

- 5.1 Limiting number of concurrent browsers on node
- 5.2 Difference between maxSession and maxInstance
- 5.3 Configuring chrome driver and IE driver exe files on grid
- 5.4 Introduction to Grid
- 5.5 Reading Nodes and Hubs, deciding number of browsers on a Node, type of browsers TestNG configurations
- 5.6 Remote WebDriver and Desired Capabilities, Running single test on single node - serially in multiple browsers, parallel on Multiple browsers
- 5.7 Running multiple tests spread across 3 nodes – one for Firefox and one for Chrome multiple browsers
- 5.8 Running multiple tests spread across 3 nodes – one for Firefox and one for Chrome
- 5.9 Prioritizing the test cases, Prioritize Interface and Custom Prioritize
- 5.10 Sharing same web driver instance among multiple tests after prioritizing them
- 5.11 Grid Coding in eclipse, Creating a Grid sample test case
- 5.12 End to end scenario building and execution on Grid -demonstration on one node, demonstration on multiple nodes on virtual machine, demonstration on multiple nodes on virtual machine and on multiple browsers

## **Unit 6: TestNG Framework**

**06**

- 6.1 Configure Eclipse with Selenium and TestNG
- 6.2 Installation of TestNG, Create First TestNG Program and Execute
- 6.3 Check reports generated by TestNG
- 6.4 Execute only failed test cases in Selenium Webdriver
- 6.5 TestNG Listeners, Implement TestNG Listener in Selenium Webdriver
- 6.6 Run group Test cases in Selenium using TestNG and Multiple Programs using TestNG
- 6.7 Execute Selenium Webdriver, Test cases parallel using TestNG
- 6.8 XSLT report generation using TestNG and Ant
- 6.9 Building a BAT file to run tests using ANT
- 6.10 Putting Data providers for multiple tests in a single file
- 6.11 Parameterizing /sharing single data provider for multiple test cases
- 6.12 TestNG Maven Configuration, Maven-surefire-report-plugin with testing, Executing TestNG from maven on Command Prompt, Generating Maven Surefire Reports and XSLT Reports
- 6.13 Managing multiple test suites

## **Unit 7: Page Object Model**

**03**

- 7.1 Concept of Page Object Model (POM)
- 7.2 Use of POM
- 7.3 Examples of POM
- 7.4 POM using Page Factory
- 7.5 Sample Examples for POM, Web Variables

**Reference Books:**

1. Srinivasan Desikan and Gopala swami Ramesh- Software Testing Principles and practices- Pearson Education India
2. Praveen -Learning Selenium IDE Automation Testing Tool Kindle Edition
3. Unmesh Gundecha, Satya Avasarala - Selenium WebDriver 3 Practical Guide: End to-end automation testing for web and mobile browsers with Selenium WebDriver.

**CA-604-MJP : Lab course based on CA-601-MJ and CA-603-MJ**  
**Sessions 15 : (Credits- 02)**

**Assignments of Full Stack Development - II:**

1. Installation and configuration of React and necessary tools.
2. Design and handle forms in React.
3. Creating a simple counter using React which increments or decrements count dynamically on-screen as the user clicks on the button.
4. Implement checklist in React. Page should display list of options, and selected choices on the web page dynamically.
5. Implement Search Filter in React.
6. Implement authentication process using JWT concepts.
7. Create a route for adding a to do item and adding dynamic routes for each to do item
8. Implement Socket programming
9. Deploying and hosting React.js Applications

**Software Testing Assignments**

1. Writing Test cases - Positive and Negative for various applications
2. Use of various components of Selenium
3. Execute Test cases in Selenium using TestNG and generate reports TestNG Assignments
4. CI/CD pipeline with Selenium
5. Configure Maven and Generate report

# CA-610-MJ : Web Services

Lectures: 30 (Credits- 02)

## Prerequisites:

- XML, HTTP, TCP/IP concepts.

## Course Objectives:

1. To Understand Web Services and implementation model for SOA.
2. To learn architecture of web services.
3. To understand cloud computing as a web service.
4. To learn WSDL.
5. To understand UDDI registry.
6. To learn restful web services.

## Course Outcomes: On completion of this course, students will be able to:

1. Understand the principles of SOA CO.
2. Efficiently use market leading environment tools to create and consume web services.
3. Identify and select the appropriate framework components in creation of web service solution.
4. Understand how to describe web services.
5. Understand register and discover web services.
6. State the difference between SOAP and Restful web services.

## Unit 1: Introduction to Web Service and SOA fundamentals

05

- 1.1 Introduction, Concept of Software as a Service (SaaS)
- 1.2 Concept and need of Web services
- 1.3 Working of Web Services
- 1.4 Web services versus Web based applications
- 1.5 Characteristics of Web services
- 1.6 Service interface and implementation
- 1.7 The Service Oriented Architecture(SOA), Quality of service (QoS)
- 1.8 Web service interoperability
- 1.9 Introduction to authorization and authentication

## Unit 2: Web Services Architecture

05

- 2.1 Web Service Architecture
- 2.2 Web service Characteristics
- 2.3 Core building blocks of web services
- 2.4 Standards and technologies available for implementing web services
- 2.5 Web services communication models
- 2.6 Basic steps of implementing web services
- 2.7 Developing web services enabled applications

<b>Unit 3: Brief Overview of XML</b>	<b>08</b>
3.1 XML Document structure, XML namespaces	
3.2 Defining structure in XML documents	
3.3 SOAP - Simple Object Access Protocol, Inter-application communication and wire protocols	
3.4 SOAP as a messaging protocol, SOAP Building Blocks, Structure of a SOAP message	
3.5 Steps of creating and consuming SOAP Web Services	
3.6 Features of SOAP, advantages and disadvantages of SOAP	

<b>Unit 4: Describing Web Services</b>	<b>04</b>
4.1 WSDL introduction	
4.2 WSDL document	
4.3 WSDL elements	
4.4 WSDL binding	
4.5 WSDL tools	
4.6 WSDL port type	
4.7 Limitations of WSDL	

<b>Unit 5: Registering and Discovering Services</b>	<b>04</b>
5.1 Concept of UDDI	
5.2 Uses and Need of UDDI Registry	
5.3 Structure of UDDI Registry	
5.4 Technical Architecture of UDDI	
5.5 UDDI Interfaces, UDDI Enquiry APIs, UDDI Publishing APIs	

<b>Unit 6: Introduction to Restful Web services</b>	<b>04</b>
6.1 Concept and need of Restful Web Services	
6.2 Restful Key Elements	
6.3 Restful Architecture	
6.4 Restful Principles and constraints	
6.5 Soap verses Restful web services	

**Reference Book:**

1. XML, Web Services, and the Data Revolution, F.P.Coyle, Pearson Education.
2. Building web Services with Java, 2nd Edition, S. Graham and others, Pearson Education.
3. Java Web Services, D.A. Chappell & T. Jewell, O'Reilly, SPD.
4. McGovern, et al., "Java web Services Architecture", Morgan Kaufmann

## CA-611-MJP : Lab course based on Web Services

No. of Sessions 15 : (Credits- 02)

### Pre-requisites:

- Strong knowledge about Java programming / PHP / .Net Framework
- Good Understanding of Object Oriented Programming concepts.
- Must be familiar with XML

### List of Assignments

1. Create 'Dynamic Web Project', which will host your web service functionality to greet the user according to server time and create 'Dynamic Web Project', which will host the client application that will send user name and test the web service Basic programs on conditional statements, looping statements.
2. Create 'Dynamic Web Project', which will host your web service functionality to convert Celsius to Fahrenheit and create 'Dynamic Web Project', which will host the client application that will send Celsius and test the web service.
3. Create 'Dynamic Web Project', which will host your web service functionality to find the factorial of given number and create 'Dynamic Web Project', which will host your web service functionality to greet the user according to server time and create 'Dynamic Web Project', which will host the client application that will send user name and test the web service Web Project', which will host the client application that will send positive integer number and test the webservice.
4. Create 'Dynamic Web Project', which will host your web service functionality to validate email id (use regular expression) and create 'Dynamic Web Project', which will host the client application that will send email id and test the web service.
5. Create 'Dynamic Web Project', which will host your web service functionality to validate user name and password (use database for storing username and password) and create 'Dynamic Web Project', which will host the client application that will send user name and password and test the web service.
6. Create 'Dynamic Web Project', which will host your web service functionality to select employee details (use database for storing emp details (eno, ename, designation, salary)) and create 'Dynamic Web Project', which will host the client application that will send employee name and display the details.
7. Create 'Dynamic Web Project', which will host your web service functionality to select Movie details (Movie(mno, mname, release\_year) and Actor(ano, aname), 1 :M cardinality ) and create 'Dynamic Web Project', which will host the client application that will send actor name and display the details.
8. Create 'Dynamic Web Project', which will host your web service functionality to validate mobile no (use regular expression: should contain only 10 numeric no) and create 'Dynamic Web Project', which will host the client application that will send mobile no and test the web service,



9. Create 'Dynamic Web Project', which will host your web service functionality to convert Rupees to Dollar, Pound, Euro, ..... and create 'Dynamic Web Project', which will host the client application that will send amount in Rupees & type of conversion and tests the web service.
10. Create 'Dynamic Web Project', which will host your web service functionality to give the suggestion for given key word and create 'Dynamic Web Project', which will host the client application that tests the web service.
11. Create 'Dynamic Web Project', which will host your web service functionality to find area and volume of the circle and create 'Dynamic Web Project', which will host the client application that tests the web service.
12. Create 'Dynamic Web Project', which will host your web service functionality to find number of vowels in the given string and create 'Dynamic Web Project', which will host the client application that tests the web service.
13. Create 'Dynamic Web Project', which will host your web service functionality to convert decimal number to Binary, Octal, Hexa Decimal and create 'Dynamic Web Project', which will host the client application that will send decimal number & type of conversion and test the web service.
14. Create 'Dynamic Web Project', which will host your web service functionality to validate user name and password (use database for storing username and password) and create 'Dynamic Web Project', which will host the client application that will send user name and password and test the web service.
15. Create 'Dynamic Web Project', which will host your web service functionality for returning book price and create 'Dynamic Web Project', which will host the client application that will send Book Name.

## CA-612-MJ : Django

Lectures: 30 (Credits- 02)

**Prerequisites:** None

### **Course Objectives:**

1. To introduce Django Framework
2. To create views and perform URL mapping
3. To create Templates and Forms in Django
4. To create Database Models and add Dynamic content to your webpages

### **Learning Outcomes: On completion of this course, students will be able to:**

1. Get acquainted with Django Framework
2. Create views with Django
3. Create Serve static content and files using Django
4. Connect templates with models to serve data dynamically
5. Create Models and understand how to connect them with Templates and Views

### **Unit 1: Introduction to Django Web Framework 02**

- 1.1 Introduction to Django Web Framework
- 1.2 Features of Django
- 1.3 Installing Django
- 1.4 MVC model
- 1.5 HTTP concepts
- 1.6 Views
- 1.7 URL Mapping

### **Unit 2: Templates and Forms 04**

- 2.1 Django Template Language
- 2.2 Utilities of Templates
- 2.3 Creating Template Objects
- 2.4 Tags, Variables and Filters
- 2.5 Rendering Templates
- 2.6 Template Inheritance
- 2.7 Form Handling
- 2.8 Form validation and Error Messages
- 2.9 Form Display

### **Unit 3: Models and Dynamic Webpages 10**

- 3.1 Django Models
- 3.2 Model Fields
- 3.3 Model Inheritance

- 3.4 CRUD on DB
- 3.5 Primary keys and the Model
- 3.6 Dynamic Webpages
- 3.7 Toggle Hidden Content
- 3.8 jQuery and AJAX integration

#### **Unit 4: Serialization**

**07**

- 4.1 Serialization and Deserialization
- 4.2 Django REST Framework
- 4.3 Serializer class
- 4.4 Model Serializers
- 4.5 REST APIs

#### **Unit 5: Parsing XML and JSON with Python**

**07**

- 5.1 Audience analysis
- 5.2 XML - RPC
- 5.3 XML, parsing object to XML and back
- 5.4 JSON, parsing object to JSON and back

#### **Reference Books:**

1. Aidas Bendoraitis, Jake Kronika, Django 3 Web Development Cookbook, Packt Publishing, Fourth Edition, ISBN-13: 978-1838987428

#### **Web References:**

1. <https://docs.djangoproject.com/en/3.2/intro/tutorial01/>
2. <https://www.djangoproject.com/start/>

## CA-613-MJP : Lab course based on Django

No. of Sessions 15 : (Credits- 02)

### List of Assignments:

1. Create a web page that outputs "Hello Django".
2. Create a Django built-in login form with password validation.
3. Design Django Admin App site in which, you can
  - Add new user.
  - Modify existing user.
  - Search and filter users.
  - Sort the user data by clicking on the column header.
  - Delete user
4. Create Django templates that represent the HTML GUI that the client can view.
5. Build Django application that illustrate template inheritance.
6. Build, handle, submit & validate HTML forms in the Django way.
7. Write a Python program to connect a database and create SQLite table within the database.
8. Write a Python program to list the tables of given SQLite database file.
9. Write a Python program to create a table and insert some records in that table. Finally selects all rows from the table and display the records.
10. Write Python Django program to insert, update and delete record in to database table using GUI.
11. Create Django app that will print records of employee having attributes first name, lastname and Designation from Django admin.
12. Build the Django form which will accept the name, email and address. Validate Name field. Name should start with 'A', else display error message.
13. Query the created models & connect to MySQL database.
14. Using Object Relational Mapper(ORM), design Django app that can insert and access the data from the student's database.
15. Create Django authentication (or auth) app that provides a wide array of tools for User management ranging from authenticating users to resetting passwords.
16. Design Django application that will create contact-us page. And after submitting information it displays Thank you message on the same page.
17. Write REST API for performing below operation to manage student record:
  - Add
  - Fetch
  - Update
  - Delete

## CA-631-RP : Research Project

(Credits- 04)

### Guidelines :

1. Students should work in a team of minimum 2 and maximum 3 students.
2. Students can choose a project topic without any restriction on technology or domain.
3. Students are expected to carry out the following tasks during project work –
  - a. Problem Identification
  - b. Literature Review/ Study
  - c. Feasibility Study
  - d. Design (includes DB design, system flow or design diagrams)
  - e. Modelling (if applicable)
4. Track sheet will be maintained by project guide for each group separately.
5. Project guide will conduct presentation for the work done (mentioned in point no. 3)
6. Project groups will work on actual development and/or implementation of proposed idea/topic.
7. Record of progress will also be maintained by keeping track sheet.
8. At the end of the project, the group should prepare a report which should conform to international academic standards. The report should follow the style in academic journals and books, with clear elements such as: abstract, background, aim, design and implementation, testing, conclusion and full references, Tables and figures should be numbered and referenced to in the report.
9. Minimum 2 demos will be conducted for the project work.
10. The final project presentation with demonstration (EE) will be evaluated.

## **SECOND YEAR : SEMESTER II**

### **CA-651-MJP : Industrial Project Design**

**(Credits- 04)**

#### **Course Objectives:**

1. To develop skills in the application of theory to practical work situations
2. To apply the design techniques studied in theory to a real life project.
3. To expose students to real work environment.

#### **Learning Outcomes:** After completion of the course, students will be able to:

1. Apply fundamental principles of software design to real world problems.

#### **Guidelines :**

1. Each student will take up either training at an industry/research institute or will work on campus on a project idea. This will be a full time activity for the period of minimum four months.
2. The institute will appoint faculty members to work as coordinators/mentors to supervise this activity.
3. Student has to follow the software development life cycle for the software project, which they are doing in industry / research institute and design the software system using UML techniques.
4. Continuous assessment will be carried out by the faculty mentor for 50% marks on the basis of weekly attendance and the correctness of the design.
5. A student will produce a hard copy of the project report at the time of final exam, which will include all the UML diagrams and student will have to explain his/her project with the help of diagrams.

## **CA-660-MJ and CA-661-MJ : MOOC courses**

**(2 courses of 2 credits each)**

### **Guidelines :**

1. Students are supposed to complete 2 online MOOC courses during their second semester of second year.
2. Students can opt for Swayam, NPTEL, Spoken tutorial courses.
3. Students should register for these courses in parallel to their Industrial Training.
4. Each course should be of minimum 30 hours (2 credits)
5. Students will have to appear for online exams of the courses and should obtain the passing certificate to earn the credits.
6. The number of hours of the course must be mentioned on the certificate and it should be minimum 30 for gaining 2 credits.
7. The courses should be relevant to Computer Science

## **CA-681-OJT : Industrial Training / Industrial Project**

**(Credits- 14)**

### **Course Objectives:**

1. To develop skills in the application of theory to practical work situations
2. To provide students the opportunity to test their interest in a particular career
3. To expose students to real work environment experience, gain knowledge in writing report in technical works/projects.
4. To build strength, teamwork spirits and self-confidence in student..

### **Learning Outcomes:** After completion of the course, students will be able to:

1. Apply fundamental principles of the subjects to solve real world problems.
2. Become master in at least one specialized area
3. Able to communicate efficiently
4. Ability to identify, formulate and model problems and find solutions.

### **Guidelines :**

1. Each student will take up either training at an industry/research institute or will work on campus on a project idea. This will be a full time activity for the period of minimum four months.
2. The institute will appoint faculty members to work as coordinators/mentors to supervise this activity.
3. Students will meet the coordinator/mentor at least once in a week and will submit synopsis and weekly report of the work done during the period of Industrial training (ITP).
4. Continuous assessment will be carried out by the faculty mentor for 50% marks on the basis of weekly attendance, performance, progress report, presentations given by the student.
5. After Completion of the ITP, a student will have to submit the project completion certificate from the respective industry/research institute.
6. A student will produce a hard copy (Hard Bound with black colour cover and Golden Embossing on it) and a soft copy of the report in the specified format.
7. End semester examination will be conducted by two examiners: faculty mentor, expert from industry (appointed by the college).
8. Student will have to present the work done by him/her in the period of Industrial Training with the help of PowerPoint presentation.



## Evaluation Pattern

### The internal and external evaluation will be 50-50%

For all the courses, which are of four credits, total marks will be 100. Out of 50 marks will be allotted for internal evaluation and 50 marks for external evaluation.

For all the courses, which are of two credits, total marks will be 50. Out of 25 marks will be allotted for internal evaluation and 25 marks for external evaluation.

#### Theory Courses of four credits :

- Internal evaluation will be of 50 marks for which 4 continuous evaluation exams of 10, 10, 10 and 20 marks will be conducted
- External evaluation will be of 50 marks

#### Theory Courses of two credits :

- Internal evaluation will be of 25 marks for which 3 continuous evaluation exams of 10, 5 and 10 marks will be conducted
- External evaluation will be of 25 marks

#### Practical Courses of four credits :

- Internal evaluation will be of 50 marks out of which 30 marks will be for assignment submissions done throughout the semester and a test/viva will be conducted for 20 marks
- External evaluation will be of 50 marks

#### Practical Courses of two credits :

- Internal evaluation will be of 25 marks out of which 15 marks will be for assignment submissions done throughout the semester and a test/viva will be conducted for 10 marks
- External evaluation will be of 25 marks

#### Methods of assessment for internal evaluation:

Seminar, objective test, open book test, Quiz, viva, projects, assignments, group discussion, research paper review, case study, industrial visit

#### Passing percentage

The student must secure at least 40% marks of that course to earn the full credit.

Examination	Credits	Marks Out of	Passing marks (40%)
Internal	4	50	20
External	4	50	20
Internal	2	25	10
External	2	25	10

**Note:** There is separate passing for internal and external examinations.