DISCIPLINE SPECIFIC CORE COURSE – 13 Full Stack Web Development 2

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Full Stack Web Development 2	4	3	0	1	Class XII Pass	DSC-10

Learning Objectives

1. Assimilate and master latest framework like frameworks like JS, Node.JS, and Mongo DB.

Learning Outcomes

1. Able to use basic to advanced Node JS.
2. Integrate Node JS with mongo database.
3. Install and use different tools like Github, Maven and Jenkins.
4. Develop a fully functioning website and deploy on a web server.

SYLLABUS OF DSC-13

Unit I Introduction to Node JS (6 Hours)

What is Node.js, Why Node.js, Node in-built packages (buffer, fs, http, os, path, util, url), Node.js Modules, Import your own Package, Node Package Manager (NPM), Local and Global Packages, File System: Get Input from Users, Pass Multiple Arguments with Yargs, File System Module.

Unit 2 Advanced Node JS (12 Hours)

Express Framework, Run a Web Server using Express Framework, Routes, Deploy application using PM2 and Nginx, Asynchronous Programming- Call Stack, Callbacks, Callback Queue and Event Loop, Callback Abstraction, Callback Chaining.

Unit 3 Integration of Node.js with Mongo DB (9 Hours)

Inserting Documents, Querying, Updating and Deleting Documents, Connect Mongo DB and Node.js Application, REST API.

Unit 4 Overview of Git, Jenkins and Maven: Git (9 Hours)

Understand the differences between Git, Github and Gitlab, Install and configure Git for use, Use Git to manage files using CLI commands, Create, Clone and manage repositories.

Jenkins- Jenkins and its architecture, Jenkins tools management, user management in Jenkins

Maven - Maven project structure, maven plugins, Project object model (POM), maven build lifecycle, adding external dependencies to maven pom.xml, maven build and test project.

Unit 5 Introduction to Docker (9 Hours)

Comparing VM and Docker, Docker- an Architectural overview, The Docker Hub A brief Introduction, Preparing docker - machine- Installation and configuration, Start containerizing, Play with docker images, Customizing container on your own, Running Container with Docker - commands, Port forwarding with docker container.

Essential Readings

1. Brad Dayley, Node.js, Mongo DB and Angular Web Development: The definitive guide to using the MEAN stack to build web applications (Developer's Library), 2nd edition, Addison-Wesley, 2018.
2. John Edward, Cooper Berg, DevOps. Building CI/CD Pipelines with Jenkins, Docker Container, AWS ECS, JDK 11, Git and Maven 3, Kindle Edition.

Practical component

A web development project implementing the technologies such as Node JS, Mongo DB, Angular JS, jQuery, JavaScript, Git, Jenkins and Maven.

Consider the following case study.

Smart Evaluation Portal is an application used for conducting online exams for the students. Exam consists of objective type questions with only 4 options, among them 1 option will be correct. Two types of Users are maintained in the application :

(i) Admin User : Admin user is responsible for creating the Exam, adding the questions and generating the results.

(ii) Student User : Student user will attempt the test that was created by the admin user in a particular time limit.

Following are the requirement map for the application as Admin Perspective:

(a) Admin can register and login with his/her email id and password.
(b) Admin can Create a New Test by giving information like Name of Test, Duration of Test, Total Marks of Test, Number of Questions in Test, Date and Time of the Test.
(c) Admin can Edit/View/Insert and delete the Questions prepared for the test.
(d) Admin can Plan multiple tests at same time by maintaining 4 status of the Tests as : Draft Tests, Scheduled Test, Result Awaited Tests, Live Tests. At a particular time, the test will be at one of the following stages.
(e) Admin Can Block, Unblock user at the time of test.

Following are the requirement map for the application as Student Perspective:

(a) Students can Register herself/himself at the time of taking the test.
(b) Students can Mark answers according to the question by choosing any one option from the given options.
(c) Students can submit the test in between as well.

Perform Following Practicals based on above requirements.

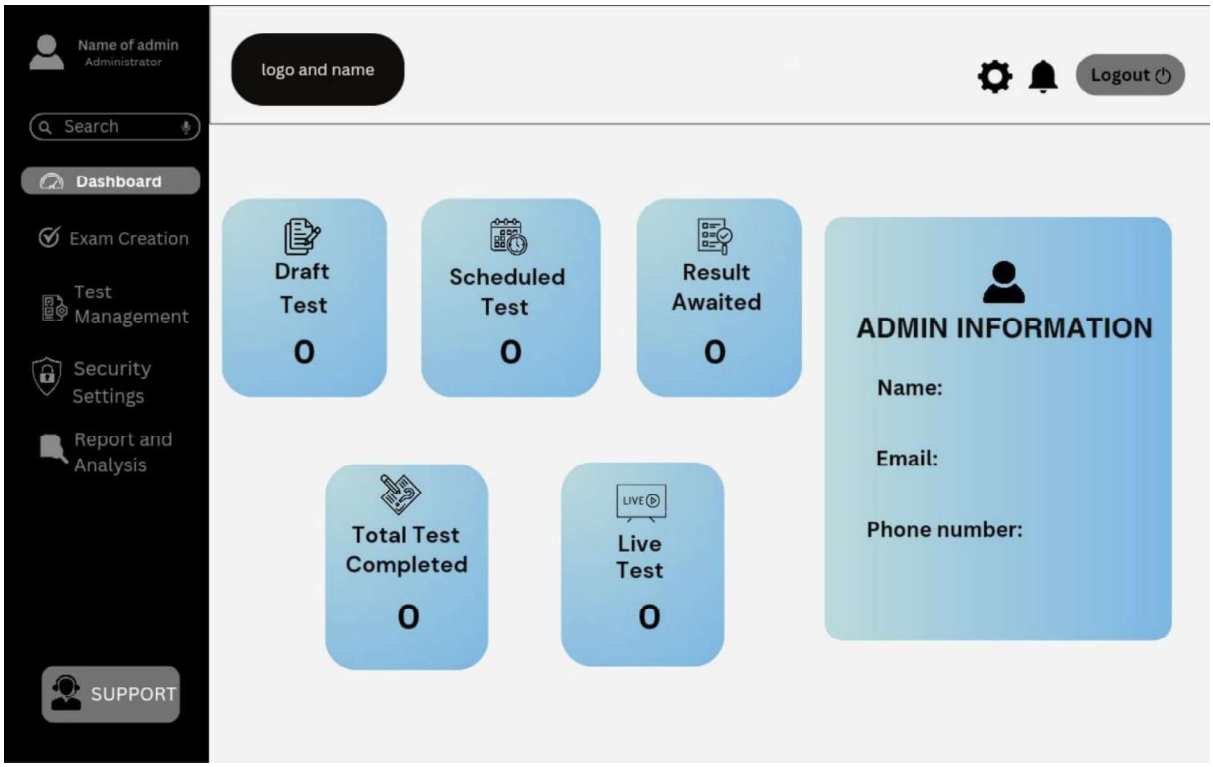
1. Configure and Install Git and Create a Git repository of the Smart Evaluation Portal application by using the following commands -

Git init
Git add <file names>
Git Commit -m “Smart Evaluation Portal”

2. Create a Github repository on Github Cloud and Push the git repository (Created in Practical 1) into that Github repository by using CLI Commands.

3. Create HTML Forms using HTML, CSS, AngularJS for the Admin Registration which includes following fields : Name, email_id, Phone number, password etc. Data must be captured using Node.js framework and data will get stored in MongoDB Database.

4. Create Admin Dashboard by using HTML, CSS, Node JS and MongoDB. Use Asynchronous Programming wherever needed in the application. Event loops and Callback Chaining must be implemented during creation of dashboards.



5. Create MongoDB Containers and Insert/Delete and Update documents in the following containers developed for Smart Evaluation Portal :

a) Admin Information
b) Exam Information
c) Test Management

6. Create a Maven Spring Boot Project with POM.xml file which will used for Inserting, updating and deleting user responses in the particular tests. Maven project will directly connect to Mongodb database using JDBC or hibernate. Add all required dependencies in POM.xml to run that project.

7, i. Install Docker into your system.
ii.Create Docker File of Smart Evaluation Portal Project
iii.Develop the docker Image for the application which consist of all the required dependencies.
iv.Upload that Docker image on Docker Hub Cloud platform.

8. Use Docker Hub image uploaded in Practical 7 and execute it in your local Machine by using CLI Docker commands and execute it to make it a docker Container.

9. Use Jenkins CI/CD Pipeline for updating and changing the new requirements in the Smart Evaluation Portal for Software maintenance and Management.

10. Use Git Commands to Commit, Push and Pull Changes from the github repository to your local machine Project. Following Commands must be used :

i) git Push Origin
ii) git Commit -m
iii) git pull

DISCIPLINE SPECIFIC CORE COURSE – 14 Software Engineering

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
Software Engineering	4	3	0	1	Class XII Pass	NA

Learning Objectives

This course will provide fundamental approaches and techniques used to develop good-quality software. This includes learning of various software development process frameworks, requirement analysis, design modeling, qualitative and quantitative software metrics, risk management, and testing techniques.

Learning Outcomes

On successful completion of the course, students will be able to:

1. Understand the software development models.
2. Analyze and model customer requirements and build design models.
3. Estimate and prepare schedule for software projects.
4. Analyze the impact of risks involved in software development.
5. Design and build test cases, and to perform software testing.

SYLLABUS OF DSC-14

Unit 1 Introduction (9 Hours)

Software Engineering - A Layered Approach; Software Process – Process Framework, Umbrella Activities; Process Models – Waterfall Model, Incremental Model, and Evolutionary process Model (Prototyping, Spiral Model); Introduction to Agile, Agile Model – Scrum.

Unit 2 Software Requirements Analysis and Specification (6 Hours)

Use Case Approach, Software Requirement Specification Document, Flow-oriented Model, Data Flow Model.

Unit 3 Design Modeling (9 Hours)

Translating the Requirements model into the Design Model, The Design Process, Design Concepts - Abstraction, Modularity and Functional Independence; Structure Charts.

Unit 4 Software Metrics and Project Estimation (9 Hours)

Function based Metrics, Software Measurement, Metrics for Software Quality; Software Project Estimation (FP based estimations); Project Scheduling (Timeline charts, tracking the schedule).

Unit 5 Quality Control and Risk Management (3 Hours)

Quality Control and Quality Assurance, Software Process Assessment and Improvement; Software Risks, Risk Identification, Risk Projection, Risk Mitigation, Monitoring and Management.

Unit 6 Software Testing (9 Hours)

Strategic Approach to Software Testing, Unit Testing, Integration Testing, Validation Testing, System Testing; Black-Box and White Box Testing, Basis Path Testing.

Essential Readings

1. R. S. Pressman, Software Engineering: A Practitioner's Approach, 9th Edition, McGraw-Hill, 2020.

2. K. K. Aggarwal, Yogesh Singh, Software Engineering, 3rd Edition, New Age International Publishers, 2007.

Suggested Readings

1. P. Jalote, An Integrated Approach to Software Engineering, 3rd Edition, Narosa Publishing House, 2005.

2. Ian Sommerville, Software Engineering, 9th Edition, Addison Wesley, 2011.

3. Ken Schwaber, Jeff Sutherland, The Definitive Guide to Scrum: The Rules of the Game, 2016.

Practical component

Create a project report that includes the following–

1. Define Problem Statement and identify process model.
2. Requirement Analysis: Create Use Case diagram and SRS Document.
3. Software Matrices and Project Estimation: Compute Function Point, estimate Effort and Cost, identify Risks, and create a Timeline Chart.
4. Software Design: Create Structured Chart.

5. Coding: Implement at least one module in a programming language of your choice.
6. Testing: Compute Cyclomatic Complexity and generate some test cases for validation.

Some of the Sample Projects are given below thought they are not limited to this.

1. College Canteen Automation System
2. Online Car-Pooling System
3. Patient Appointment System
4. Online Attendance Management System
5. Medical Prescription Processing System

DISCIPLINE SPECIFIC CORE COURSE – 15 WordPress and Framework

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
WordPress and Framework	4	3	0	1	Class XII Pass	DSC-11, DSC-12

Learning Objectives

The objective of this course is to motivate student's interest in web application framework for building scalable web application development.

Learning Outcomes

On successful completion of the course, students will be able to:

1. Identify basic aspects of web-frameworks.
2. Apply the basic concepts, principles and practices of Web-site development using server-side technologies (PHP & MySQL).
3. Create and manage Blogs,Websites using WordPress.
4. Usage of PHP & MySQL with WordPress.

SYLLABUS OF DSC-15

Unit 1: Web Technologies (9 Hours)

HTML, CSS, Java Script, Ajax. Server side scripting language – PHP, Relational Database Management System - MySQL

Unit 2: Introduction to Web Frameworks (3 Hours)

Introduction to WordPress Web Frameworks, Working of WordPress.

Unit 3: Introduction to WordPress (9 Hours)

Introduction to Blogging, Creating Blogs, Using Images, Wrapping Text Around

Images, Comments, Post Formats, Linking to Posts, Pages, and Categories, Using Smilies, Links Manager, WordPress Feeds, Customizing Feeds, Use Gravatars in WordPress, Writing Code in Your Posts, Using Password Protection.

Unit 4: Design features of WordPress (21 Hours)

Developing a Colour Scheme, Designing Headers, CSS Horizontal Menus, Dynamic Menu Highlighting, Navigation Links, Next and Previous Links, Styling for Print, Designing Your Post Meta Data Section, Separating Categories in your Post Meta Data Section, Customizing the Read More, Formatting Date and Time, Finding CSS Styles, Creating Individual Pages, Uploading Files, Using WordPress Themes, Templates, Template Tags, Template Hierarchy, Validating a Website, Know the Sources, WordPress Site Maintenance.

Unit 5: WordPress with MySQL (3 Hours)

Working of MySQL in WordPress.

Essential Readings

1. Brian Messenlehner, Jason Coleman, Building Web Apps with WordPress: WordPress as an Application Framework, 2nd Edition, O'Reilly, 2019.
2. Jeff Siarto, Head First WordPress: A Brain-Friendly Guide to Creating Your Own Custom WordPress Blog, O'Reilly, 2010.
3. http://www.wpbeginner.com/beginners-guide/how-to-learn-wordpress-for-free-in-aweek-or-less/

Practical component

Create a blog website using WordPress, focusing on the following aspects:

1. **Setup and Configuration:**
    i. Install WordPress on a hosting platform or local server.
    ii. Configure a MySQL database and integrate it with WordPress.
2. **Theme Customization and Design:**
    i. Select a WordPress theme or develop a custom theme.
    ii. Customize the theme extensively by:
        - Creating a unique color scheme that complements your brand.

- Designing headers with graphical elements or sliders.
- Implementing CSS for horizontal menus with dynamic highlighting.
- Styling for print and ensuring readability across different devices.
- Designing post meta data sections and formatting date/time displays.

3. **Content Creation and Management:**
   i. Create and publish a variety of blog posts (minimum 4) showcasing different:
      - Post formats (e.g., standard, image, video, gallery).
      - Use of images with text wrapping and alignment.
      - Integration of Gravatars and customizing comment sections.
      - Password-protecting selected posts for restricted access.
4. **Enhanced Functionality with Plugins:**
   i. Install and configure essential WordPress plugins to enhance functionality, such as:
      - SEO optimization plugin for better search engine visibility.
      - Social media sharing and integration plugins.
      - Advanced custom fields for additional content flexibility.
      - Contact form plugin with Ajax submission for seamless user interaction.
5. **WordPress Features Implementation:**
   i. Utilize WordPress core features effectively:
      - Linking posts, pages, and categories for seamless navigation.
      - Managing comments and integrating social media links.
      - Customizing RSS feeds and ensuring compatibility with feed readers.