DISCIPLINE SPECIFIC CORE COURSE – 10 Full Stack Web Development -1

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 -1	4	3	0	1	Class XII Pass	Introduction to Web Programming (DSC-05)

Learning Objectives

1. To introduce the fundamentals of the Internet, and the principles of web design.
2. To construct basic websites using JQuery and AJAX.

Learning Outcomes

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

1. Assimilate and master the latest framework like frameworks like js, Node.js, and MongoDB.
2. Build a Responsive Web application using Angular Typescript
3. Implement Angular Binding and events with templates
4. Write queries in Mongo database.
5. Implement Mongo databases and formulate queries for data retrieval and data update problems.

SYLLABUS OF DSC-10

Unit 1 Introduction to JQuery (12 hours)

JQuery Introduction, JQuery Syntax, JQuery Selectors, JQuery Events, JQuery Effects- JQuery Hide/Show, JQuery Fade, JQuery Slide(), JQuery Animate, JQuery Stop(), JQuery Callback, JQuery Chaining, JQuery AJAX- JQuery AJAX Introduction, JQuery Load, JQuery Get/Post, JQuery HTML, JQuery Get, JQuery Set, JQuery Add, JQuery Remove, JQuery CSS Classes, JQuery CSS(), JQuery forms.

Unit 2 Introduction to Angular JS (6 hours)

Angular Architecture, Building blocks of Angular, Angular CLI and commands, Angular Modules, Understanding files in Angular, Angular forms.

Unit 3 Working of Angular Applications (9 hours)

Angular App Bootstrapping ,Angular Components, Creating A Component Through Angular CLI , Ways to specify selectors , Template and styles , Installing bootstrap to design application , Data Binding , Types of Data Binding , Component Interaction using @Input and @Output decorator , Angular Animations , Component Life-cycle Hooks , Angular Directives.

Unit 4 Introduction of Mongo DB (9 hours)

Overview , Design Goals for Mongo DB Server and Database, Mongo DB Tools , How to modularize code by separating routes , Usage of various Mongo DB Tools available with Mongo DB Package , Mongo DB Development Architecture.

Unit 5 Crud Operations (9 hours)

Mongo DB CRUD Introduction, Mongo DB Datatypes , Analogy between RDBMS & Mongo DB Data Model, Mongo DB Data Model (Embedding & Linking), Challenges for Data Modelling in Mongo DB.

Practical component

Experiment1:

Write a JQuery program to retrieve element Id's & classes of HTML elements of below program.

```
<!DOCTYPE html>
<html>
<head>
<style>
/* Style the element with the id "myHeader" */
#myHeader {
  background-color: lightblue;
```

```
  color: black;
  padding: 40px;
  text-align: center;
}

/* Style all elements with the class name "city" */
.city {
  background-color: tomato;
  color: white;
  padding: 10px;
}
</style>
</head>
<body>

<h2>Difference Between Class and ID</h2>
<p>A class name can be used by multiple HTML elements, while an id name must only be used by one HTML element within the page:</p>

<!-- An element with a unique id -->
<h1 id="myHeader">My Cities</h1>

<!-- Multiple elements with same class -->
<h2 class="city">London</h2>
<p>London is the capital of England.</p>

<h2 class="city">Paris</h2>
<p>Paris is the capital of France.</p>

<h2 class="city">Tokyo</h2>
<p>Tokyo is the capital of Japan.</p>
```

```
</body>
</html>
```

Experiment 2:

Program for Selectors in jquery?

1. On click of a button, welcome message should display.
2. There are five input textboxes. On click of the input box , the background colour should change to yellow. (Hint : use *this*)
3. Remove and add different style to the element by using JQuery. Create two style classes and by default apply first one to one element. On click of a button change the elements associated property to other style.
4. Toggle the style properties of the element on click
5. Create one button and a textbox. On click of the button the text written in text box should display on the button.

Experiment 3:

Program to use show(), hide() and toggle() ?

1. Create a div tag and it should display Welcome message (by using JQuery)
2. Create two div tags and assign yellow as background colour to both of them by using JQuery.
3. Display and hide message shown in the div tag on click of the buttons. You can use Jquery show hide.
4. Display and hide message by using a single button . You can use Jquery toggle().
5. Create a div tag and on click of a button it should display your name.

Experiment 4:

1. Create one style class and name it *error*, give its border colour as red and background colour as yellow. Apply this class to the text box when the focus is moved out of it and if entered data is more than 10.
2. Create two textboxes and one button with + on it. On click of the button you should display sum of the two entered numbers in textbox. Similarly add Subtract, Multiple and Division buttons.
3. Your buttons in above code will remain in disable state till both the inputs are filled. (Use one function to enable or disable buttons. Trigger that function with on Blur events of the textboxes)

Experiment 5:

Program to use GET and POST Methods in jquery form ?

1. Create a form asking users to enter userid and password. On submit of the form the page should submit the data to backend page written in PHP or Python. Note that you are submitting the form using JQuery so page should not reload or redirect. If submitted data is valid then the login form should show welcome message by hiding the input boxes, otherwise it should ask the user to try again.
2. You have created one style css error in previou exercise above. Use the same when wrong user id or password is entered.

Experiment 6:

Perform these angular command line Interface commands to create following angular components.

1. Command to create angular project
2. Command to create a new component
3. Command to create a new class in angular project
4. Command to create a new directive
5. Command to create a new module
6. Command to create a new angular Service

Experiment 7:

Create an Angular form (Template driven Angular form) to register a student in an university with following fields.

a) Students name
b) Students Roll No
c) Students Address
d) Students email Id
e) Students Contact Number

Input data must be printed on the console after clicking on the submitted button.

Experiment 8:

Create an Angular Form (Reactive driven Angular form) to login a user in any application with these input fields.

a) Login Id (Email Id)

b) Password (Length constraints)

Input data must be printed on the console after clicking on the submitted button.

Format of MongoDB Query

Structure of 'restaurants' collection:


```
{
 "address": {
    "building": "1007",
    "coord": [ -73.856077, 40.848447 ],
    "street": "Morris Park Ave",
    "zipcode": "10462"
 },
 "borough": "Bronx",
 "cuisine": "Bakery",
 "grades": [
    { "date": { "$date": 1393804800000 }, "grade": "A", "score": 2 },
    { "date": { "$date": 1378857600000 }, "grade": "A", "score": 6 },
    { "date": { "$date": 1358985600000 }, "grade": "A", "score": 10 },
    { "date": { "$date": 1322006400000 }, "grade": "A", "score": 9 },
    { "date": { "$date": 1299715200000 }, "grade": "B", "score": 14 }
 ],
 "name": "Morris Park Bake Shop",
 "restaurant_id": "30075445"
}
```

Based on the above format perform the below experiments.

Experiment 9:

a) Write a MongoDB query to display the fields restaurant_id, name, borough and cuisine for all the documents in the collection restaurant.

b) Write a MongoDB query to display the fields restaurant_id, name, borough and cuisine, but exclude the field _id for all the documents in the collection restaurant.

c) Write a MongoDB query to display all the restaurant which is in the borough Bronx.

d) Write a MongoDB query to display the first 5 restaurant which is in the borough Bronx.

Experiment 10:

a) Write a MongoDB query to display the next 5 restaurants after skipping first 5 which are in the borough Bronx.

b) Write a MongoDB query to find the restaurants who achieved a score more than 90.

c) Write a MongoDB query to find the restaurants that achieved a score, more than 80 but less than 100.

d) Write a MongoDB query to find the restaurants which locate in latitude value less than -95.754168.

Experiment 11:

a) Write a MongoDB query to find the restaurants which do not prepare any cuisine of 'American' and achieved a grade point 'A' not belongs to the borough Brooklyn. The document must be displayed according to the cuisine in descending order.

b) Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which contain 'Wil' as first three letters for its name.

c) Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which contain 'ces' as last three letters for its name.

d) Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which contain 'Reg' as three letters somewhere in its name.

Experiment 12:

Create a Full Stack Web application Portal using Angular JS and Mongo DB.

Application Name – College Feedback Application

Technology used – Angular JS and Mongo DB

Users – Faculty, Students, Administrator

Requirements – Faculty can develop or create a Feedback form from predefined fields.

Faculty can filter/choose the students list as per the requirements.

Faculty can set the time limit to post the feedback.

Students can find the feedback lists for those they are authorized.

Students can submit the feedback.

Data must be stored as Key-Value pair in Mongo DB and use HTML, CSS and Angular JS for user interface.

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. Cody Lindley, JQuery Cookbook, O'Reilly, 2009.

DISCIPLINE SPECIFIC CORE COURSE – 11 PHP Programming

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

Learning Objectives

This course is designed to develop an understanding of the functionality and utility of PHP scripts and the usage of syntax, variables, and datatypes available in PHP. It provides the necessary knowledge to create basic web applications using PHP script.

Learning Outcomes

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

1. Implement the installation and deployment of PHP.
2. Develop simple programming constructs in PHP.
3. Design and develop simple, dynamic web applications using PHP scripts.

SYLLABUS OF DSC-11

Unit 1 Introduction to PHP (6 hours)

PHP introduction, inventions, and versions, important tools and software requirements (like Web Server, Database, Editors etc.), PHP with other technologies, the scope of PHP, Basic Syntax, PHP variables and constants, Types of data in PHP, Expressions, scopes of a variable (local, global), PHP Operators: Arithmetic, Assignment, Relational, Logical operators, Bitwise, ternary Grade A Grade B Grade C Grade and MOD operator, PHP operator Precedence and associativity.

Unit 2 Handling HTML form with PHP (6 hours)

Capturing Form Data, GET and POST form methods Dealing with multi-value fields Redirecting a form after submission

Unit 3 PHP conditional events and Loops (6 hours)

PHP IF Else conditional statements (Nested IF and Else) Switch case, while, For and Do While Loop, Goto, Break, Continue, and exit.

Unit 4 PHP Functions (9 hours)

Function, Need of Function, declaration and calling of a function, PHP Function with arguments, Default Arguments in Function, Function argument with call by value, call by reference, Scope of Function Global and Local.

Unit 5 String Manipulation and Regular Expression (9 hours)

Creating and accessing String, Searching & Replacing String, Formatting, Joining and Splitting String, String Related Library functions, Use and advantage of regular expression over inbuilt function, Use of preg_match(), preg_replace(), preg_split() functions in the regular expression.

Unit 6 Array (9 hours)

Anatomy of an Array, Creating index-based and Associative array, Accessing array Looping with Index-based array, with associative array using each() and foreach(), Some useful Library function.

Practical component

Experiment1:

Write a simple program to check the eligibility of a candidate in India to caste a vote (on the basis of Age).

Experiment 2:

Use PHP operators to find the cube root and square root of a given number n. If the number is not a perfect cube/square then return its floor value.

Experiment 3:

Write a PHP script to print the below following pattern.

a)

```
      1
   1  0  1
1  0  1  0  1
```

Experiment 4:

Write a PHP script to find the Highest common factor, GCD of given two numbers a and b.

Experiment 5:

Create a HTML Form to register a student in a university with the following fields:

f) Students name
g) Students Roll No
h) Students Address
i) Students email Id
j) Students Contact Number

The fields should be submitted to students.php file with request type:

1) POST

2)GET

Experiment 6:

Write a PHP script to Sort an Array of objects of object fields.

Experiment 7:

Write a PHP program to find the factorial of a number using while loop, for loop and do while loop.

Experiment 8:

Write a program to find the Fibonacci series of a number N by using

a) Recursive Approach
b) Iterative Approach

Experiment 9:

You need to write a PHP script that will convert convert temperatures between Celsius & Fahrenheit. The input parameters should be the temperature value and the unit (Celsius or Fahrenheit). The script will return the convert value in the other unit.

Experiment 10:

Create a PHP script that shows a list of products in a simple e-commerce catalog. Each product will have a name, description, price and featured image. You can store the product data in an array, a JSON file or a database. The script should generate an HTML page that will display the product details in a user-friendly format.

Experiment 11:

Write a PHP script to validate emailID and phone no. using regular expression using "preg_match()".

Experiment 12:

An English paragraph is given

"#Video provides a powerful way to help you prove your point, When you click Online Video, you can paste in the embed code for the video you want to add. You can also type @ keyword to search online for the video th@t best fit$ your document.

*## To make your document look professionally produced, Word provide$ header, footer, cover p@ge, and text box designs that complement each other. For ^example,% you can add a matching cover page, **header, and sidebar. Click Insert and then choose the elements you want from the different galleries%."*

Write a PHP program to remove all the special characters, convert all uppercase to lowercase, replace space with tab using PHP String Functions and regular expression functions.

Experiment 13:

Write a PHP script to rotate an array by K elements.

Experiment 14:

Write a PHP script to check whether the given numbers *a* and *b* are Anagrams or not (Two integers are considered to be digit anagrams if they contain the same digit or one can be obtained from the other by rearranging it).

Experiment 15:

Create an associative array to store marks in five subjects of a student, where key is subject name and value is marks in the subject. Display all keys and values. Also sort the array on the basis of the key.

Experiment 16:

Find all the leader in the given array. Leader is defined as the number after which all the numbers are less than that.

Example :

$myArr = 2 8 3 6 4 1

Output : Leaders are – 1,4,6,8

Experiment 17 :

Write a program in PHP Script where a Indexed based array and a number K is given. Check whether an array contains the two numbers whose sum is equal to K.

Example : $myArr = 3,7,6,9,2 K = 9

Output = True (3 & 6)

Experiment 18:

Write PHP script to

a) Create a database "CollegeWebsite"
b) Create a table "User (username varchar(120), password varchar(10))" within database "CollegeWebsite".
c) Create a login page, which asks the user for a username and password. On clicking submit, a welcome message should be displayed if the user is already registered (i.e., name is present in the database) otherwise an error message should be displayed.

.

Essential readings

1. Steven Holzner, PHP: The Complete Reference, 1st Edition, McGraw Hill Education, 2007.
2. Timothy Boronczyk, Martin E. Psinas, PHP and MYSQL (Create-Modify-Reuse), Wiley India Private Limited, 2008.
3. Robin Nixon, Learning PHP, MySQL, JavaScript, CSS & HTML5, 3rd Edition, O'Reilly, 2014.
4. Luke Welling, Laura Thompson, PHP and MySQL Web Development, 4th Edition, Addison-Wesley, 2008.

DISCIPLINE SPECIFIC CORE COURSE – 12 DBMS using MySQL

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

Learning Objectives

The course introduces the students to the fundamentals of database management system and its architecture. Emphasis is given on the popular relational database system including data models and data manipulation. Students will learn about the importance of database structure and its designing using conceptual approach using Entity Relationship Model and formal approach using Normalization. The importance of file indexing and controlled execution of transactions will be taught. The course would give students hands-on practice of structured query language in a relational database management system and glimpse of basic database administration commands.

Learning Outcomes

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

1. Use database management system software to create and manipulate the database.
2. Create conceptual data models using entity relationship diagrams for modeling real-life situations and design the respective database schema.
3. Use the concept of functional dependencies to remove redundancy and update anomalies.
4. Apply normalization theory to get a normalized database scheme.
5. Write queries using relational algebra.
6. Implement relational databases and formulate queries to get the desired solutions for a broad range of data retrieval and data update problems using MySQL.

SYLLABUS OF DSC-12

Unit 1 Introduction to Database (6 hours)

Purpose of database system, Characteristics of database approach, data models, database management system, database system architecture, three-schema architecture, components of DBMS, data independence, and file system approach vs database system approach.

Unit 2 Entity-Relationship Modeling (6 hours)

Conceptual data modeling - motivation, entities, entity types, attributes, relationships, relationship types, constraints on relationship, Entity Relationship diagram notation.

Unit 3 Relational Data Model (9 hours)

Update anomalies, Relational Data Model - Concept of relations, schema-instance distinction, keys, relational integrity constraints, referential integrity and foreign keys, relational algebra operators and queries.

Unit 4 Structured Query Language (SQL) (9 hours)

Querying in SQL, DDL to create database and tables, table constraints, update database-update behaviors, DML, aggregation functions, group by and having clauses, retrieve data from the database, generate and query views. Access and manipulate databases using ODBC. Basic Database administration SQL commands.

Unit 5 Database Design (9 hours)

Mapping an Entity Relationship model to the relational database, functional dependencies and Normal forms, 1NF, 2NF, 3NF and BCNF decompositions and desirable properties of them.

Unit 6 File Indexing and Transaction Processing (6 hours)

Data Storage and Indexes- Need of file indexes, file organizations, index structures, single- and multi-level indexing, concurrent execution of transactions, ACID properties, need of data recovery and log file.

Practical component

Practical based on MySQL.

1. Create a table EMPLOYEE with following schema:

(**Emp_no**, E_name, E_address, E_ph_no, Dept_no, Dept_name,Job_id(char) , Salary)

Write SQL statements for following queries:

i. Add a new column; Hire_Date to the existing relation.

ii. Change the datatype of Job_id from char to varchar2.

iii. Change the name of column Emp_no to E_no.

2. Create a table named EMPNEW with the following structure:

EMPNO NUMBER (5)

ENAME VARCHAR2 (20)

JOB VARCHAR2 (10)

DEPTNO NUMBER (3)

SALARY NUMBER (7,2)

Write SQL statements for following queries:

i. Allow NULL for all columns except EMPNO,ENAME and JOB.

ii. Add constraints to check, while entering the EMPNO > 100.

iii. Define the field DEPTNO as unique.

iv. Create a primary key constraint for the table (EMPNO).

v. Create a foreign key constraint on the EMPNEW table's attribute EMPNO, referencing the EMPLOYEE table's Emp_no column.

3. Create a table DUEMPLOYEE with following schema:

*(**Emp_no**, E_name, E_address, E_ph_no, Dept_no, Dept_name,Job_id , Salary)*

Write SQL statements for following queries:

i. Insert atleast 5 rows in the table.

ii. Display all the information of DUEMPLOYEE table.

iii. Display the record of each employee who works in department CS.

iv. Update the city of Emp_no-2 with current city as New Delhi.

v. Delete the email_id of employee whose Emp_no is 5.

4. Create a table INDUSTRYEMPLOYEE with following schema:

*(**Emp_no**, E_name, E_address, E_ph_no, Dept_no, Dept_name,Job_id, Designation,Date_of_Joining, Salary)*

Write SQL statements for the following queries.

i. List the Emp_no, E_name, Salary of all employees working as MANAGER.

ii. Display all the details of the employees whose salary is more than the Salary of any SOFTWARE ENGINEER.

iii. List the employees in the ascending order of Designations of those joined after 2023.

iv. List the E_name those are starting with 'A'.

v. List all the employees except VICE PRESIDENT in asc order of Salaries.

5. Create a table DEPARTMENT with following schema:

*(**Dept_no**, Dept_name,No_of_Emp)*

Write SQL statements for the following queries.

i. Display all the Dept_no available with the DEPARTMENT and INDUSTRYEMPLOYEEtables avoiding duplicates.

ii. Display all the Dept_no available with the DEPARTMENT and INDUSTRYEMPLOYEEtables.

iii. Display all the Dept_no available in INDUSTRYEMPLOYEE table and not in DEPARTMENT table and vice versa.

6. Consider the following schemas:

Sailors(**sid**,sname,rating,age)

Boats(**bid**,bname,color)

Reserves(**sid,bid**,day(date))

Write SQL statements for the following queries.

i. Find all information of sailors who have reserved bid 101.

ii. Find the name of boat reserved by Arun.

iii. Find the names of sailors who have reserved a red boat, and list in the order of age.

iv. Find the names of sailors who have reserved at least one boat.

v. Find the ids of sailors who have reserved a red boat or a green boat.

vi. Find the name and the age of the youngest sailor.

vii. Count the number of different sailor names.

viii. Find the average age of sailors for each rating level.

7. Create a table CSINDUSTRYEMPLOYEE with following schema: *(**Emp_no**,*

E_name, E_address, E_ph_no, Dept_no, Dept_name,Job_id,

Designation,Date_of_Joining, Salary,Manager_name)

Write SQL statements for the following queries.

i. Display total salary spent for each Job_id.

ii. Display lowest paid employee details under each manager.

iii. Display number of employees working in each department and their department name.

iv. Display the details of employees according to salary in decreasing order.

v. Show the record of employee earning salary greater than 50000 in each department.

Essential Readings

1. Elmasri Ramez, Shamkant B. Navathe, Fundamentals of Database Systems, 7th Edition, Pearson Education, 2015.
2. Jesper Wisborg Krogh, MySQL Connector/Python Revealed: SQL and NoSQL Data Storage Using MySQL for Python Programmers, Apress, 2018.
3. Joel Murach, Murach's MySQL, 3rd edition, Pearson, 2019.

Suggested Readings

1. Raghu Ramakrishnan, Johannes Gehrke, Database Management Systems, 3rd Edition, McGraw-Hill, 2014.
2. Abraham Silberschatz, Henry F. Korth, S. Sudarshan, Database System Concepts, 7th Edition, McGraw Hill, 2021.
3. Thomas Connolly, Carolyn Begg, Database Systems: A Practical Approach to Design, Implementation, and Management, 6th edition, Pearson, 2021.