INSTITUTE	**FACULTY OF TECHNOLOGY**
PROGRAM	**BACHELOR OF TECHNOLOGY (INFORMATION & COMMUNICATION TECHNOLOGY)**
SEMESTER	**6**
COURSE TITLE	**ADVANCED WEB TECHNOLOGIES**
COURSE CODE	**01CT1625**
COURSE CREDITS	**4**

Objective:

1 The objective of the course is to understand modern web technologies based on JavaScript.

Course Outcomes: After completion of this course, student will be able to:

1 Implement NoSQL Database CRUD operations
2 Acquire knowledge about Server-side JS framework to make Database Connectivity
3 Acquire knowledge about functionalities of Client-side and Server-side JS frameworks
4 Explore ReactJS features and create component-based web pages
5 Develop Front-end web pages and connect to the Back-end Databases

Pre-requisite of course: Student should have a basic understanding of fundamental languages used by the browser and the server. Student should be confident in designing and styling web pages using HTML & CSS. Fundamental concepts of JavaScript should be clear and should also be familiar with DOM(Document Object Model) in browser. Basic understanding of Node/express will speed up the learning process but is not recommended.

Teaching and Examination Scheme

Theory Hours	Tutorial Hours	Practical Hours	ESE	IA	CSE	Viva	Term Work
3	0	2	50	30	20	25	25

Contents : Unit	Topics	Contact Hours
1	**Client-side JS Framework – ReactJS** Introduction to ReactJS, Introducing JSX, Rendering Elements, Component and Props, State and Life Cycle, Handling Events, Conditional Rendering, List and Keys, Forms, Lifting State Up, Composition vs Inheritance, Thinking in React, Manual and Unit Testing	12
2	**Server-side JS Framework – Node JS** Get started with Node.js, Node Package Manager, Modules Asynchronous Program, Callbacks, Events and Event Loop, Streams and Buffers, Connect Node.js to Database, Web Sockets, API using NodeJS	5

MR. CHANDRASINH DHIRUBHA PARMAR
Digitally signed by (Name of HOD)

DR. RAJENDRASINH BAHADURSINH JADEJA
Digitally signed by (Name of Dean/ Principal)

Contents : Unit	Topics	Contact Hours
3	**Express JS** MVC Pattern, Introduction to Express, Routing, HTTP Interaction, Handle Form Data, Handle Query Parameters, Cookies and Sessions, User Authentication, Error Handle, Create and Consume RESTful Services, Use Templates, Functional and Unit Testing	12
4	**MongoDB** Introduction to MongoDB, CRUD Operations in Mongodb, Introduction to Mongoose, Core concepts of Mongoose, Extending Models - Working with hooks, Validation of model data, Creating custom static methods, Creating custom instance methods, CRUD operations with Mongoose	5
5	**Cloud & Deployment** Basics of Devops and Cloud Infrastructure, In-depth of DNS and network terminologies, Introduction to AWS, AWS-ECS, EC2, Docker, Kubernetes, Git Basics, Git with CI/CD, Hosting website on various platforms.	8
Total Hours		**42**

Suggested List of Experiments:

Contents : Unit	Topics	Contact Hours
1	**Experiment-1** Create an application for student result management using React JS.	2
2	**Experiment-2** Develop a web application which involves database operations using NodeJS	2
3	**Experiment-3** Improve any React (+NodeJS) application by adding Error Handling.	2
4	**Experiment-4** Develop a web application which involves database operations using Express	2
5	**Experiment-5** Create modern, scalable and high-speed Web Applications with React and Node.js +Express	2
6	**Experiment-6** Create a RESTful service with Node, Express and MongoDB	2
7	**Experiment-7** Frontend: Using ReactJS, make Image Carousel, FAQ/Accordion and Shopping item list with filters.	2
8	**Experiment-8** Frontend: Using ReactJS, integrate video player like a youtube clone	2

Suggested List of Experiments:

Contents : Unit	Topics	Contact Hours
9	**Experiment-9** Frontend: Using ReactJS, make a e-commerce website (flipkart or amazon) clone	2
10	**Experiment-10** Backend: Using NodeJS, build a real time chat application or email sender.	2
11	**Experiment-11** Backend: Using NodeJS, build a job search platform or e-commerce platform api to search and filter.	2
12	**Experiment-12** Database: Using MongoDB, build the CRUD operations with NodeJS	2
13	**Experiment-13** Database: using MongoDB, NodeJS and ReactJS, build a project to fetch and post data from frontend to database.	2
14	**Experiment-14** Deployment: Using AWS, deploy the project in EC2 instances or use Lambda functionalities to build APIs.	2
Total Hours		**28**

Textbook :

1 Node.js by Example, Krasimir Tsonev, Packt Publishing Ltd, 2015

2 Fullstack React: The Complete Guide to ReactJS and Friends, Anthony Accomazzo, Fullstack.io, 2017

References:

1 MongoDB Cookbook , MongoDB Cookbook , Amol Nayak, Packt Publishing, 2016

Suggested Theory Distribution:

The suggested theory distribution as per Bloom's taxonomy is as follows. This distribution serves as guidelines for teachers and students to achieve effective teaching-learning process

Distribution of Theory for course delivery and evaluation					
Remember / Knowledge	**Understand**	**Apply**	**Analyze**	**Evaluate**	**Higher order Thinking / Creative**
15.00	30.00	20.00	15.00	10.00	10.00

Instructional Method:

1 The internal evaluation will be done on the basis of continuous evaluation of students in the laboratory and class-room.

MR. CHANDRASINH DHIRUBHA PARMAR
Digitally signed by (Name of HOD)

DR. RAJENDRASINH BAHADURSINH JADEJA
Digitally signed by (Name of Dean/ Principal)

Instructional Method:

2 Practical examination will be conducted at the end of the semester for evaluation of performance of students in laboratory.

3 Students may use supplementary resources such as online videos, NPTEL videos, e-courses, Virtual Laboratory, etc.

4 The course delivery method will depend upon the requirement of content and need of the students. The teacher in addition to conventional teaching method (Chalk and Talk) may use any of the tools such as demonstration, role play, Quiz, brainstorming, Flipped class, Project based learning, Collaborative learning, MOOCs etc. for effective teaching.

Supplementary Resources:

1 https://reactjs.org/docs/getting-started.html

2 https://nodejs.dev/en/learn/