RVS COLLEGE OF ARTS AND SCIENCE

Autonomous and Affiliated to Bharathiar University, Approved by AICTE

Re Accredited with ‘A+’ Grade by NAAC

Sulur, Coimbatore – 641 402.

Estd 1986

RVS CAS

Building Intellectual Capital

SCHOOL OF COMPUTER STUDIES

MASTER OF COMPUTER APPLICATIONS

(Effective from the Academic Year 2023- 2024)

HOD **PRINCIPAL** **COE**

VISION

Quality Education for Digital Era.

MISSION

To impart a need – based quality education through comprehensive curriculum by adopting apt technologies and progressive teaching, learning and research processes.

ABOUT THE DEPARTMENT

The Department of Master of Computer Applications (MCA) is one of the pioneer Departments in the College and was started in the year 1989. Since then under the eminent guidance of the Management, it has grown by producing more than 30 batches of MCA Students from the College. Under its roof are dynamic, well-qualified, experienced teaching and non-teaching members of Staff. Close to 123 students are studying in the MCA Programme.

Towards the vision of the College, the School of Computer Studies (MCA) currently has incorporated Full Stack Development skills in the Curriculum. These are the most important skills that are in demand today.

In future, there will be enormous growth potential. Towards this, RVSCAS School of Computer Studies has collaborated with Transform Tech Pvt. Ltd., and has launched Academic Garage.

Academic Garage:

A pioneer classroom in India where academia and industry shake hands with state-of-the-art pedagogy through start-up environment experience. The students are benefited as the practical and project based learning has replaced traditional learning. The students are evaluated by Hackathon Assessment Method through real-time project development. The successful graduates meet the promising interviewers with their real-time projects and not by their resume.

The School provides world class curriculum with the help of internationally certified faculty and skilled industrial experts.

EXECUTIVE MEMBERS

CHAIRMAN

Dr. K.V. Kupusamy

MANAGING TRUSTEE

Dr. K. Senthil Ganesh

CORRESPONDENT

Mrs. S. Srividyalakshmi Senthil Ganesh

SECRETARY

Prof. Saramma Samuel

PRINCIPAL

Dr. T. Sivakumar

VICE PRINCIPAL

Dr. M.P. Ayyappadas

CONTROLLER OF EXAMIATIONS

Ms. G. Jeyalakshmi

DIRECTOR- MCA

Dr. P. Navaneetham

PROGRAMME OUTCOMES (POs)

PO1	Domain Knowledge
PO2	Communicative Competence
PO3	Digital Strategic Knowledge
PO4	Multi-Cultural Competence
PO5	Critical Thinking and Problem Solving
PO6	Research and Analytical Skills
PO7	Moral, Ethical and Professional Responsibilities
PO8	Leadership and Lifelong Learning

PROGRAMME SPECIFIC OUTCOMES (PSOs)

PSO1	To perform the Job Roles such as Full Stack Web & Mobile Developer, Full Stack Developer(MERN), Node/React Developer, Web Developer, React Native Developer.
PSO2	To Apply the skill sets of HTML, CSS, JavaScript and Bootstrap.
PSO3	To develop web applications using MongoDB, ExpressJS, ReactJS and NodeJS.
PSO4	To develop mobile applications using React NativeJS, NodeJS and MongoDB.

RVS COLLEGE OF ARTS AND SCIENCE (Autonomous)
Sulur, Coimbatore – 641 402
SCHEME OF EXAMINATIONS

MASTER OF COMPUTER APPLICATIONS(MCA)
2023- 2025 BATCH

Semester	Course Opted	Course Name	D	L	T	P	FA	SA	Marks	Credits
SEMESTER I	**M-I**	HTML & CSS	6	4	2	-	25	75	100	6
	M-II	Web Development with JavaScript	3	3	2	6	25	75	100	8
	MP-I	Practical-I: Web Designing using HTML & CSS	3	-	-	6	25	75	100	3
	PV	Mini Project in Web Development with JavaScript	3	-	5	6	25	75	100	8
	MANDATORY	Career and Professional Development - I	3	2	-	-	25	75	100	1
	Total			**36**					**500**	**26**

Semester	Course Opted	Course Name	D	L	T	P	FA	SA	Marks	Credits
SEMESTER II	**M-III**	Back End Development using Node JS and MongoDB	6	4	2	-	25	75	100	8
	MP-II	Practical-II: Server Side Programming using Node JS	3	-	-	6	25	75	100	6
	EL-I	Elective-I	3	3	2	6	25	75	100	3
	PV	Mini Project in Web Development with MERN Stack	3	-	5	6	25	75	100	8
	MANDATORY	Career and Professional Development - II	3	2	-	-	25	75	100	1
	Total			**36**					**500**	**26**

Semester	Course Opted	Course Name	D	L	T	P	FA	SA	Marks	Credits
SEMESTER III	**M-IV**	Full Stack Development Tools	3	3	2	6	25	75	100	8
	MP-III	Practical-III: Mobile App Development	3	-	-	6	25	75	100	6
	EL-II	Elective-II	6	4	2	6	25	75	100	3
	PV	Mini Project in Mobile App Development using React Native	3	-	5	6	25	75	100	8
	MANDATORY	Career and Professional Development - III	3				25	75	100	1
	Total			**30**						**26**

Semester	Course Opted	Course Name	D	L	T	P	CIA	SEE	Marks	Credits
SEMESTER IV	**PV**	Major Project and Viva Voce	-	-	0	30	25	75	100	15
	Total					**30**			**500**	**15**
***Electives** **# Project & Viva voce** **$ Mandatory**									**700+200* + 400# +300$**	
TOTAL									**1600**	**93**

M-Major Paper; **MP**-Major Practical

ELECTIVE I-	**ELECTIVE II-**
Front End Development with React	Mobile App Development using React Native
Front End Development with Angular JS,	Mobile App Development using Flutter
Front End Development with Vue JS	Mobile App Development using Android

Career and Professional Development

Students have to successfully complete the mandatory Extra Credit Course in Career and Professional Development I-III to get 3 Credits.

SEMESTER-I

MAJOR-1

Course Title: **HTML & CSS (T)**	Course Code : **13P**
Semester : **I**	Course Group : **M**
Teaching Scheme in Hrs (L:T:P) : **4:2:0**	Credits : **6**
Map Code: **NA**	Total Contact Hours : **90**
CIA : **25 Marks**	#SEE: **75 Marks**
Programme: **MCA**	# - Semester End Exam

No.	**Course Outcome**	**PSOs**	**Cl. Ses**	**CL**
CO1	Illustrate the basic concepts of web and HTML file structure	PSO1, PSO2	12L + 6T	AP
CO2	Apply style to the static webpage, box model, page layout ,lists and links	PSO1, PSO2	12L + 6T	AP
CO3	Demonstrate the use of images in web pages	PSO1, PSO2	12L + 6T	AP
CO4	Compute table layout in web pages	PSO1, PSO2	12L + 6T	AP
CO5	Implement form controls, audio and video in web pages	PSO1, PSO2	12L + 6T	AP

PEDAGOGY:

Lecture, Group Discussion, LCD, Seminar and Case Study.

UNIT – I **18 Hours**

Introduction to web development: How web applications work **(**The components of a web application, How static web pages are processed, How dynamic web pages are processed, A survey of web browsers and server-side scripting languages, How client-Side JavaScript fits into web development) - An introduction to HTML and CSS **(**The HTML for a web page, The CSS for a web page, A short history of the HTML and CSS standards) - Tools for web development (Text editors for HTML and CSS, IDEs for web development, FTP programs for uploading files to the web server) - How to view deployed web pages (How to view a web page, How to view the source code for a web page) - Five critical web development issues **(**Users and usability, Cross-browser compatibility, User accessibility, Search engine optimization, Responsive Web Design)

How to code, test, and validate a web page: The HTML syntax (The basic structure of an HTML document, How to code elements and tags, How to code attributes, How to code comments and white space) The CSS syntax (How to code CSS rule sets and comments, How to code basic selectors) - How to use VS Code to work with HTML and CSS files (How to create a project, How to open an HTML file, How to start a new HTML file, How to edit an HTML file, How to open or start a CSS file, How to edit a CSS file, How to preview and run an HTML file) - How to test, debug, and validate HTML and CSS files **(**How to test and debug a web page, How to validate an HTML file, How to validate a CSS file).

How to use HTML to structure a web page : How to code the head section (How to code the title element, How to link to a favicon, How to include metadata) - How to code text elements **(** How to code headings and paragraphs, How to code special blocks of text, How to code inline elements for formatting and identifying text, How to code character entities, How to code the core attributes) - How to structure the content of a page (How to use the primary HTML5 semantic elements, How to use some of the other HTML5 semantic elements, When and how to code div and span elements) - How to code links, lists, and images (How to code absolute and relative URLs, How to code links, How to code lists, How to include images) - A structured web page **(**The page layout, The HTML file)

UNIT – II **18 Hours**

How to use CSS to format the elements of a web page: An introduction to CSS(Three ways to provide CSS styles for a web page, Two ways to provide for browser compatibility, How to specify measurements and colors, How to specify measurements, How to specify colors, How to use the CSS3 color specifications) - How to code selectors(How to code selectors for all elements, element types, ids, and classes, How to code relational selectors, How to code combinations of selectors, How to code attribute selectors, How to code pseudo-class and pseudo-element selectors) - How to work with Cascading Style Sheets(How the cascade rules work, How to use the developer tools to inspect the styles that have been applied) - How to work with text **(**How to set the font family and font size, How to set the other properties for styling fonts, How to set properties for formatting text, How to use CSS3 to add shadows to text, How to float an image so text flows around it) - A web page that uses external style sheets (The page layout, The HTML file, The CSS file).

How to use the CSS box model for spacing, borders and backgrounds: An introduction to the box model **(**How the box model works, A web page that illustrates the box mode) - How to size and space elements**(**How to set heights and widths, How to set margins, How to set padding) - A web page that illustrates sizing and spacing(The HTML for the web page, The CSS for the web page, A version of the CSS that uses a reset selector) - How to set borders and backgrounds(How to set borders, How to use CSS3 to add rounded corners and shadows to borders, How to set background colors and images, How to use CSS3 to set background gradients) - A web page that uses borders and backgrounds (The HTML for the web page, The CSS for the web page)

UNIT – III **18 Hours**

How to use CSS for page layout: How to float elements in 2- and 3-column layouts(How to float and clear elements, How to use floating in a 2-column, fixed-width layout, How to use floating in a 2-column,fluid layout, How to use floating in a 3-column,fixed-width layout) - Two web pages that use a 2-column, fixed-width layout **(**The home page, The HTML for the home page, The CSS for the home page, The speaker page, The HTML for the speaker page, The CSS for the speaker page) - How to use CSS3 to create text columns(The CSS3 properties for creating text columns, A 2-column web page with a 2-column article) - How to position elements(Four ways to position an element, How to use absolute positioning, How to use fixed positioning, A table of contents that uses positioning).

How to work with lists and links: How to code lists(How to code unordered lists, How to code ordered lists, How to code nested lists, How to code description lists) - How to format lists (How to change the bullets for an unordered list, How to change the numbering system for an ordered list, How to change the alignment of list items) - How to code links (How to link to another page, How to format links, How to use a link to open a new browser window or tab, How to create and link to placeholders, How to link to a media file, How to create email, phone, and Skype links) - How to create navigation menus (How to create a vertical navigation menu, How to create a horizontal navigation menu, How to create a 2-tier navigation menu, How to create a 3-tier navigation menu, The CSS for a 3-tier navigation menu**)**

UNIT – IV **18 Hours**

How to work with images: Basic skills for working with images(Types of images for the Web, How to include an image on a page, How to resize an image, How to align an image vertically, How to float an image) - Advanced skills for working with images **(**How to use the HTML5 figure and figcaption elements, How to work with thumbnails, How to do image rollovers, How to create image maps) - Related skills for working with images (When to use an image editor, How to get images and icons, How to create favicons)

How to work with tables: Basic HTML skills for coding tables (An introduction to tables, How to create a table, How to add a header and footer) - Basic CSS skills for formatting tables **(**How to use CSS properties to format a table, How to use the CSS3 structural pseudo-classes for formatting tables) - Other skills for working with tables **(** How to use the HTML5 figure and figcaption elements with tables, How to merge cells in a column or row, How to provide for accessibility, How to nest tables, How to control wrapping).

UNIT – V **18 Hours**

How to work with forms: How to use forms and controls **(** How to create a form, How to use buttons, How to use text fields, How to use radio buttons and check boxes, How to use drop-down lists, How to use list boxes, How to use text areas, How to use labels, How to group controls with field set and legend elements, How to use a file upload control) - Other skills for working with forms **(**How to align controls, How to format controls, How to set the tab order and assign access keys) - How to use the HTML5 features for data validation (The HTML5 attributes and CSS selectors for data validation, How to use regular expressions for data validation, How to use a data list to present entry options) - How to use the HTML5 controls **(**How to use the email, url, and tel controls, How to use the number and range controls, How to use the date and time controls, How to use the search control for a search function, How to use the color control, How to use the output element to display output data, How to use the progress and meter elements to display output data) - A web page that uses HTML5 data validation(The page layout, The HTML, The CSS).

How to add audio and video to your website: An introduction to media on the web (Common media types for video and audio, Video codecs, Audio codecs, Audio and video support in current browsers, How to encode media) - How to add audio and video to a web page (**How** to use the object and param elements, How to use the embed element, How to use the HTML video and audio elements, How to fall back to Flash for backward compatibility) - A web page that offers both audio and video (The page layout, The HTML)

REFERENCES:

1. Murach's HTML5 and CSS3 (3rd Edition) by Zak Ruvalcaba and Anne Boehm.
2. HTML & CSS: The Complete Reference, Publication-McGraw Hill Education; 5th edition
by Thomas Powell 2017.
3. Mastering Responsive Web Design: Push your HTML and CSS skills to the limit and build professional grade, responsive websites BY Ricardo Zea Publication-Packt Publishing Limited - 2015.
4. Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics, Fifth Edition, 2018, by Jennifer Robbins Publication- Shroff/O'Reilly;
5. Mastering HTML, CSS & Javascript Web Publishing by Laura Lemay (Author), Rafe Colburn (Author), Jennifer Kyrnin (Author) Publication- BPB Publications; First edition-2016.

SEMESTER-I

MAJOR-2

Course Title: **Web Development with JavaScript(P)**	Course Code : **13Q**
Semester : **I**	Course Group : **M**
Teaching Scheme in Hrs (L:T:P) : **3:2:6**	Credits : **8**
Map Code: **NA**	Total Contact Hours : **165**
CIA : **25 Marks**	#SEE : **75 Marks**
Programme: **MCA**	# - Semester End Exam

No.	**Course Outcome**	**PSOs**	**Cl. Ses**	**CL**
CO1	Demonstrate key web programming concepts	PSO1, PSO2	9L + 6T + 18P	AP
CO2	Make web applications using JavaScript, HTML, and CSS	PSO1, PSO2	9L + 6T + 18P	AP
CO3	Implement user interactions on web pages	PSO1, PSO2	9L + 6T + 18P	AP
CO4	Implement UI components for web applications	PSO1, PSO2	9L + 6T + 18P	AP
CO5	Use code management tools such as GitHub	PSO1, PSO2	9L + 6T + 18P	AP

PEDAGOGY:

Lecture, Group Discussion, LCD, Seminar and Case Study.

UNIT – I **33 Hours**

Computational Thinking: Introduction To Computing(What Is A Computer And Restaurant Example)- Getting Started With HTML(Create A Web Document Using HTML)- Working with Styles(Creating Shapes In HTML And Shapes Exercise)- Bootstrap Styles(Introduction To Bootstrap)- Variables in JavaScript(Variables And Variables Code Exercise)- Arrays(Arrays And Arrays Code Exercise)- Conditional Statements(Conditional Statements And Conditional Statements Exercise)- Loops(Loops And Loops Exercise)- Functions- Functions: Working with Libraries(Functions - Libraries, Code And Loops Exercise)- Functions: Arrays(Functions - Array Exercise)- Functions : Objects(Functions : Objects Exercise)- Computational Thinking(Computational Thinking Summary)- Simulation(Simulation - Scheduling Computation)- Scheduling Repeating Computation(Simulation - Scheduling Repeating Computation)- Maintaining State- Moving in Time and Space- One Dimensional Motion and Animation- Moving In Time And Space: Two Dimensional Motion(Edge Detection And 2D animation Edge Detection)- Coding Challenge(Bonus Exercise).

UNIT – II **33 Hours**

Mental Model of Computing Operations: Mental Model Of A Computer- Thinking Through The Mental Model Of A Computer(Points To Remember When Coding)- Basic Data Types(Basic Data Types In JavaScript)- Scope Of Variables(Scope Of Variables In JavaScript)- Equals Operator- Passing Primitive Types Into Functions(Primitive Types: Pass By Value Into Function)- Passing Objects And Arrays Into Functions(Objects And Arrays Passed By Objects Into Function And Passed By Reference)- Passing Functions Into Other Functions- Passing Functions Into Other Functions(Pass And Fire Function)- Casting Types- Function Declaration Vs. Function Expression(Model Factory And Hoisting Functions And Vars)- Debugging In The Console To Understand Code Execution(Debugging In Chrome)- Debugging Basics- Debugging Exercises- Graphics Animation Of Projectiles And Random Walk Exercise- Big Bang Exercise.

Introduction to JavaScript: Variables (Introduction to Variables and Programming in JavaScript)- Arrays and Objects (First Array and First Object)- Block Scope (Block Scope of Let)- Function Calls with Primitive Types Vs. Objects for Arguments (Function Call with Argument and Function Call with Object for Argument)- Passing Arguments To Functions By Value And By Reference(Pass By Value - Pass By Reference)-Basic Looping(Basic Looping And Looping On Array Exercise)- Looping to Add Elements(Looping On Array Exercise)- PacMan Exercise(Introduction To PacMan Exercise And PacMan Exercise)-

UNIT – III **33 Hours**

Functions: Array Manipulation and Scope: Array Manipulation (Introduction to Array and String Manipulation And Array Manipulation Functions)- String Manipulation (String Manipulation Functions And String Template Literals)- Anonymous Functions (Anonymous Functions And Fat Arrows)- Function and Block Scope (Scope Of Variables)- Passing Functions by Reference- ES6 Modules (ES6 Module Demo And Module Pattern)- Walk Boston Exercise (Introduction To Walk Boston Exercise And Boston City Data)

Callback Functions: Callback Functions (Callbacks Introduction and Callback)- For Each Element in an Array (For Each Element in An Array and For Each Exercise)- Filter Callback (Filter Callback and Filter Exercise)- Sort Callback (Sort Exercise and Sort and Chart Salaries)- Map Callback- Reduce Function (Reduce)- Word Count Exercise

UNIT – IV **33 Hours**

Introduction To GitHub, Testing, And the DOM: GitHub Install with Keys (GitHub Install)- Introduction to The GitHub Cycle (Introduction To The GitHub Cycle and GitHub Cycle)- VS Code GitHub Integration- Introduction to Testing (Introduction To Testing And What Is Testing?)- Installing Node.js- Testing Hello World (Testing The Hello World Exercise)- The World Wide Web And Tim Berners Lee- Introduction to The Document Object Model- Injecting JavaScript into HTML Web Pages- The Document Object Model- Dynamically Inject Posts into Div- Render Using Fetch- PacMen Factory Exercise (Introduction To The PacMen Exercise And Factory For PacMen)-

Styles And Bootstrap: Introduction to Styles- HTML, CSS, And JavaScript (Three Languages: HTML, CSS, And JavaScript, Separating HTML, CSS, And JavaScript Into Different Files)- Styling with Class (Styling With Class, CSS - Inheritance, Selectors, and What Matters The Most (At Least, to Your Browser))- Controlling the Layout Using CSS Grid (Control Layout Using CSS Grid And Understanding Grid Lines)- Holy Grail (Holy Grail, Resources)- Bootstrap Styles-

Styling Fonts (Font Basics And Web Fonts)- Applying Styles Programmatically to Create Dynamic Pages (Applying Styles Programmatically)- Coding Challenge: Styling Programmatically (Styling The Grid Programmatically And Animated Style Application, Bringing It All Together)- Eye Movement Exercise- Working with CSS Reflection (Solutions).

UNIT – V **33 Hours**

Asynchronous Code: Asynchronous Code (Introduction to Asynchronous Code and What is Asynchronous Code?)- Promises- Async and Await- Async in The Browser (Async in The Modern Browser And Performance Budget)- Mapping Exercise (Map Hello World And Map Markers)- Mapping Visualizations (Map Clustering And Heat Maps)-Map Animation (Map Animation And Real-Time Bus Tracker)-**Introduction to Cyber Security and Recursion:** Agile Methodology-Introduction to Cybersecurity-What Is Open SSL? -Cybersecurity - Encryption, Kerberos, And PKI & PKI Keys-Hash Demo Blockies and Create Your Own Blockie-Introduction to Recursion-Permute String and Permute Exercise-Rotate Image-Rotate Matrix Exercise- Tower of Hanoi Exercise.

REFERENCES

1. Full Stack Development with MERN- MIT
2. JavaScript for Kids: A Playful Introduction to Programming | December 12, 2014|Nick Morgan
3. JavaScript Patterns: Build Better Applications with Coding and Design Patterns | First Edition| Stoyan Stefanov
4. Eloquent JavaScript, 3rd Edition: A Modern Introduction to Programming | Third Edition | Marjin Haverbeke
5. https://javascriptweekly.com/
6. https://2020.stateofjs.com/en-US/
7. https://github.com/lydiahallie/javascript-questions#readme
8. https://edabit.com/challenges/javascript

SEMESTER-I

MAJOR-3

Course Title :**PRACTICAL – I : WEB DESIGNING USING HTML & CSS (P)**	Course Code : **13S**
Semester : **I**	Course Group : **MP**
Teaching Scheme in Hrs (L:T:P) : **0:0:6**	Credits : **3**
Map Code: **NA**	Total Contact Hours : **90**
CIA : **25 Marks**	SEE: **75 Marks**
Programme: **MCA**	# - Semester End Exam

No.	**Course Outcome**	**PSOs**	**Cl. Ses**	**CL**
CO1	Illustrate the basic concepts of web and HTML file structure	PO1,PSO2	18P	AP
CO2	Apply style to the static webpage, box model, page layout ,lists and links	PO2,PSO2	18P	AP
CO3	Demonstrate the use of images in web pages	PO1,PSO2	18P	AP
CO4	Compute table layout in web pages	PO1,PSO2	18P	AP
CO5	Implement form controls, audio and video in web pages	PO1,PSO2	18P	AP

PEDAGOGY:

Lecture, Group Discussion, LCD, Seminar and Case Study.

UNIT-I **18 Hours**

HTML: Introduction to HTML (Base html page for E commerce landing page)

UNIT-II **18 Hours**

HTML: Intermediate HTML (Add table to list features in ecommerce landing page)

UNIT-III **18 Hours**

CSS: Introduction and Intermediate to CSS (Attach CSS to set font sizes, color and alignment for sections and paragraphs)

UNIT-IV **18 Hours**

Bootstrap 4: Introduction to Bootstrap 4 (add bootstrap classes like jumbotron, navbar, modal, forms, input group, collapse etc to make website appear user friendly in mobile screen.)

UNIT-V **18 Hours**

Bootstrap 4: Intermediate Bootstrap (Make sliding ads in landing pages with carousel, offer items as cards and different sizes for different orientations with media query)

REFERENCES:

1. https://www.udemy.com/the-complete-web-development-bootcamp.
2. Zak Ruvalcaba and Anne Boehm, Murach, HTML5 and CSS3, MurachPublications,4th Edition 2018.
3. Jake Spurlock, Bootstrap, O&#39; Reilly Publications, 1 st Edition2013.
4. Ben Frain, Responsive Web Design with HTML5 and CSS3, Packt Publishing, 1st edition,2012.
5. Crespo, Responsive Web Design with JQuery, Packt Publishing,1st edition 2012.

SEMESTER-I

MAJOR-4

Course Title : **Mini Project in Web Development with JavaScript**	Course Code : **13R**
Semester : **I**	Course Group : **PV**
Teaching Scheme in Hrs (L:T:P) : **0:5:6**	Credits : **8**
Map Code: **NA**	Total Contact Hours : **165**
CIA : **25 Marks**	#SEE : **75 Marks**
Programme: **MCA**	# - Semester End Exam

No.	**Course Outcome**	**PSOs**	**Cl.Ses**	**CL**
CO1	Make web applications using JavaScript, HTML and CSS	PSO1, PSO2	15L+15T +15P	AP
CO2	Implement user interactions on webpages	PSO1, PSO2	15L+15T +15P	AP
CO3	Implement UI Components for web applications	PSO1, PSO2	15L+15T +15P	AP
CO4	Apply modern software development lifecycle processes	PSO1, PSO2	5L+5T+5 P	AP
CO5	Use code management tools such As GitHub	PSO1, PSO2	5L+5T+5 P	AP

Project Submission Requirements

Use the template, which you will upload for submission. On the template you should include links to:

- Homepage
- CSS file for GitHub pages
- The file where you imported Bootstrap, for extra credit
- PacMen Exercise Repository
- Eye Exercise Repository
- Real Time Bus Tracking Repository

Website Requirements

Your website should include the following:

1. **Video Introduction**, which should be:

- An introduction to yourself, including an overview of your work and skills Uploaded onto YouTube
- Embedded on the home page of your site
- 2 minutes or less

2. Bio, which should include:
   - Minimum of 50words
   - Photo
   - GitHub username
   - LinkedIn link
   - Best way to contact you

3. Pages and Navigation, so your website includes:
   - Landing page, which contains your video introduction and bio
   - Projects page, which includes information on your coding projects
   - Navigation bar, so users can navigate between the two pages

4. Styles, you should style your website including:
   - Font
   - Color
   - Navigation
   - Image
   - Styled links
   - Use of grid layout, which will be assessed by looking at the link to your CSS file

5. Projects Page, should include the following for two or more projects. These links should go directly to the repositories you will submit for grading in the GitHub Repository Assignment Submission.
   - Project titles for the PacMen Exercise, Eye Exercise, and Real Time Bus Tracking projects
   - Links to GitHub repositories
   - Project descriptions(1-5sentencesperproject)

6. Extra Credit: Bootstrap: If you use Bootstrap for styling, you will receive extra credit. Upload a direct link to the file where you imported Bootstrap for this project to receive credit.

GitHub Repository Requirements

You will also be assessed on the **GitHub repositories** for three coding projects: PacMen Exercise, Eye Exercise, and Real Time Bus Tracker.

The **README** for each repository should contain:

1. Project Title
2. Description of Project(minimumof20words)

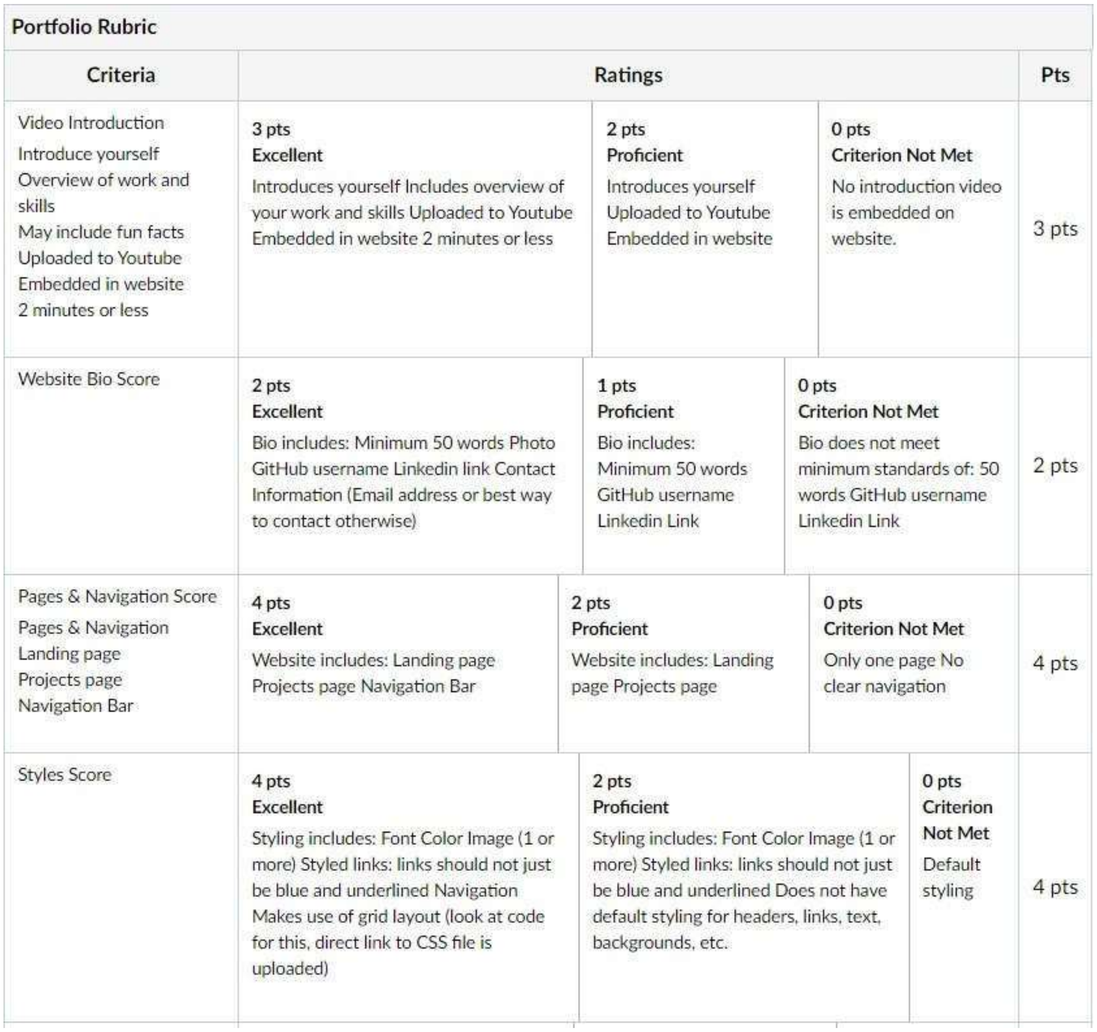
How to Run: This is information someone would use to get started running your project on their machine.

1. Road map of future improvements: This should be a few sentences describing what you plan to work on next in this repository.
2. License information: This should include information about the MIT license.

The **File structure** for each repository should align to the following:

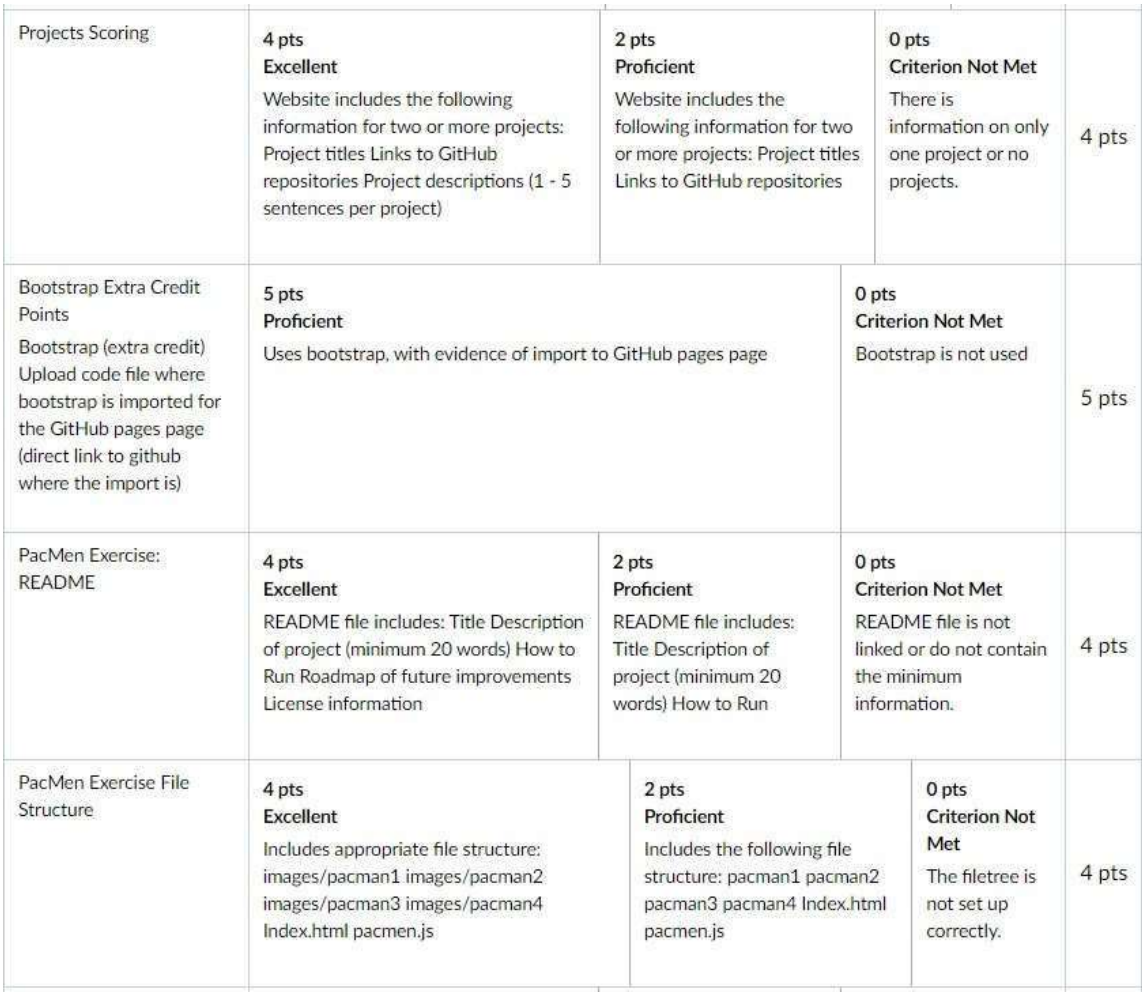
1. PacMen Exercise
   - images/pacman1
   - images/pacman2
   - images/pacman3
   - images/pacman4
   - Index.html
   - pacmen.js
2. Eye Exercise
   - index.html
   - eyes.js
   - styles.css
3. Real Time Bus Tracker
   - Index.html
   - mapanimation.js
   - styles.css

You may also review the rubric below for more information on how your work will be assessed.



Portfolio Rubric

Criteria	Ratings			Pts
Video Introduction Introduce yourself Overview of work and skills May include fun facts Uploaded to Youtube Embedded in website 2 minutes or less	**3 pts** **Excellent** Introduces yourself Includes overview of your work and skills Uploaded to Youtube Embedded in website 2 minutes or less	**2 pts** **Proficient** Introduces yourself Uploaded to Youtube Embedded in website	**0 pts** **Criterion Not Met** No introduction video is embedded on website.	3 pts
Website Bio Score	**2 pts** **Excellent** Bio includes: Minimum 50 words Photo GitHub username Linkedin link Contact Information (Email address or best way to contact otherwise)	**1 pts** **Proficient** Bio includes: Minimum 50 words GitHub username Linkedin Link	**0 pts** **Criterion Not Met** Bio does not meet minimum standards of: 50 words GitHub username Linkedin Link	2 pts
Pages & Navigation Score Pages & Navigation Landing page Projects page Navigation Bar	**4 pts** **Excellent** Website includes: Landing page Projects page Navigation Bar	**2 pts** **Proficient** Website includes: Landing page Projects page	**0 pts** **Criterion Not Met** Only one page No clear navigation	4 pts
Styles Score	**4 pts** **Excellent** Styling includes: Font Color Image (1 or more) Styled links: links should not just be blue and underlined Navigation Makes use of grid layout (look at code for this, direct link to CSS file is uploaded)	**2 pts** **Proficient** Styling includes: Font Color Image (1 or more) Styled links: links should not just be blue and underlined Does not have default styling for headers, links, text, backgrounds, etc.	**0 pts** **Criterion Not Met** Default styling	4 pts



Projects Scoring	**4 pts** **Excellent** Website includes the following information for two or more projects: Project titles Links to GitHub repositories Project descriptions (1 - 5 sentences per project)	**2 pts** **Proficient** Website includes the following information for two or more projects: Project titles Links to GitHub repositories	**0 pts** **Criterion Not Met** There is information on only one project or no projects.	4 pts
Bootstrap Extra Credit Points Bootstrap (extra credit) Upload code file where bootstrap is imported for the GitHub pages page (direct link to github where the import is)	**5 pts** **Proficient** Uses bootstrap, with evidence of import to GitHub pages page		**0 pts** **Criterion Not Met** Bootstrap is not used	5 pts
PacMen Exercise: README	**4 pts** **Excellent** README file includes: Title Description of project (minimum 20 words) How to Run Roadmap of future improvements License information	**2 pts** **Proficient** README file includes: Title Description of project (minimum 20 words) How to Run	**0 pts** **Criterion Not Met** README file is not linked or do not contain the minimum information.	4 pts
PacMen Exercise File Structure	**4 pts** **Excellent** Includes appropriate file structure: images/pacman1 images/pacman2 images/pacman3 images/pacman4 Index.html pacmen.js	**2 pts** **Proficient** Includes the following file structure: pacman1 pacman2 pacman3 pacman4 Index.html pacmen.js	**0 pts** **Criterion Not Met** The filetree is not set up correctly.	4 pts

REFERENCES

1. Full Stack Development with MERN- MIT
2. JavaScript for Kids: A Playful Introduction to Programming | December 12, 2014|Nick Morgan
3. JavaScript Patterns: Build Better Applications with Coding and Design Patterns | First Edition| Stoyan Stefanov
4. Eloquent JavaScript, 3rd Edition: A Modern Introduction to Programming | Third Edition | Marjin Haverbeke
5. https://javascriptweekly.com/
6. https://2020.stateofjs.com/en-US/
7. https://github.com/lydiahallie/javascript-questions#readme
8. https://edabit.com/challenges/javascript

SEMESTER-I

Mandatory

Course Title : **Career and Professional Development I**	Course Code : 13E
Semester : **I**	Course Group : **PG**
Teaching Scheme in Hrs (L:T:P) : **2 :0:0**	Credits : 1
Map Code : O	Total Contact Hours : **30**
CIA : **25 Marks**	SEE # : **75**
Programme: **MCA**	# - Semester End Exam

Course Outcomes (COs)

No.	After completing this course, the students will be able to	PSOs	Cl. Ses	CL
CO1	Carry themselves with grace, maintain good posture, and exhibit open and engaging body language. Build rapport, convey enthusiasm, interest, and professionalism, all of which are important for creating a positive first impression.	PO7	6	Apply
CO2	Develop and refine their communication skills, actively participate in discussions, demonstrate leadership and initiative, enhance problem-solving and decision-making abilities, strengthen interpersonal skills and teamwork, and showcase confidence and adaptability.	PO2	6	Apply
CO3	Develop effective communication skills, build confidence and self-assurance, improve clarity and articulation, and learn strategies to overcome nervousness.	PO2	6	Apply
CO4	Provide an effective and interesting overview of their qualifications, highlight relevant experiences, and demonstrate their enthusiasm for the applied job position in about one minute.	PO7	6	Create
CO5	Prepare for a professional appearance, confidence, effective verbal communication, showcasing relevant skills and experiences, and asking thoughtful questions thereby contributing to a positive and memorable interview experience.	PO5	6	Apply

UNIT: I **LECTURE HOURS: 6**

Orientation & ELT (Entry Level Test **):** Introduction about the training program and Entry Level Test

GROOMING: **<u>Internal Grooming</u> - (1)** The connection between self-confidence and grooming **(2)** Groom to Position & Present Yourself Well **(3)** Grooming Habits & Routine

<u>External Grooming</u> - (4) How to Groom Yourself **(5)** Dressing Right for a Job Interview **(6)** Building a Self-Brand

UNIT: II **LECTURE HOURS:6**

BODY LANGUAGE: (1) Discovering How Body Language Conveys Message **(2)** Hand Gestures - How the hands talk **(3)** Facial Expression - The Magic of Smile **(4)** The Head Nod **(5)** Ways to Improve your Body language **(6)** Job Interview: Making a Great First Impression

UNIT: III **LECTURE HOURS:6**

GROUP DISCUSSION: (1) Characteristics of a Group Discussion **(2**) Do's and Don'ts of GD **(3)** What you will be judged on in a GD **(4)** How to INITIATE, PARTICIPATE & CONCLUDE **(5)** How to Prepare for GD **(6)** Practical Challenges faced during a GD

UNIT: IV **LECTURE HOURS:6**

JUST A MINUTE (JAM):(1) Importance & Objective of JAM in an Interview Process. **(2)** 3 Rules of Success in JAM **(3)** 4 Key Elements of Success in JAM **(4)** Tips on Creating JAM Content **(5)** Preparing to Deliver JAM **(6)** Practical Challenges faced during JAM

UNIT: V **LECTURE HOURS:6**

SELF - INTRODUCTION: (1) Before You Give Your Self - Introduction: Understanding Self - Introduction **(2)** Rules to follow during Self - Introduction **(3)** Template to Prepare Self - Introduction: First 4 Questions **(4)** Template to Prepare Self - Introduction: Next 4 Questions **(5)** Putting it all together: Creating an impactful Self - Introduction **(6)** How to introduce yourself confidently

Reference Books:

- G. Vidyashankar, Sweep through your Interviews
- Barun K Mitra, Personality Development and Soft Skills, Oxford University Press, 2016

Assessment pattern : Practical

INTERNAL ASSESSMENT (Marks: 25)

Splitup of Marks:
1. Quiz: 5
2. Model Exam: 10
3. Assignment: 5
4. Attendance: 5

END OF SEMESTER EXAM (Practical) Marks: 75

Splitup of Marks:
PART A: Online test (MCQ) + 1 Essay type question: **20 Marks**
PART B: Practical: **40 Marks**
(a) Group Discussion: 20 Marks
(b) Self-Introduction: 10 Marks
(c) JAM (Impromptu Speech): 10 Marks
PART C: Record (Bring out your learnings in the last semester + Final Self-Introduction): **15 Marks**

SEMESTER-II

MAJOR-1

Course Title: **Elective I- Front End Development with React**	Course Code : **23P**
Semester : **II**	Course Group : **EL**
Teaching Scheme in Hrs (L:T:P) : **3:2:6**	Credits : **8**
Map Code: **NA**	Total Contact Hours : **165**
CIA : **25 Marks**	SEE # : **75**
Programme: **MCA**	# - Semester End Exam

No.	Course Outcome	PSOs	Cl. Ses	CL
CO1	Explain key React concepts such as JSX and Hooks	PSO1, PSO3	9L + 6T + 18P	AP
CO2	Design UI components using the React framework	PSO1, PSO3	9L + 6T + 18P	AP
CO3	Build UI components using the React framework	PSO1, PSO3	9L + 6T + 18P	AP
CO4	Build a React application	PSO1, PSO3	9L + 6T + 18P	AP
CO5	Test and Deploy a React application	PSO1, PSO3	9L + 6T + 18P	AP

PEDAGOGY:

- Lecture, Group Discussion, LCD, Seminar & Case Study.

UNIT – I **33 Hours**

Introduction to JavaScript ES6: Introduction To React(Getting Started With React For Front-End Development)- Introduction To ES6 And Default Parameters(Introduction To ES6)- Template Literals(Template Literals - Multiline Strings And Template Literals, Embedding Expressions)- Destructuring(Destructuring Assignment, Destructuring Assignment - Functions)- Spread And Rest(Spread And Rest And Spread - Value And Reference, Spread - Merge Arrays, Enhanced Object Literals)- Asynchronous Functions And Modules(Arrow Functions, Modules In ES6)- Var, Let, And Const (Function Scope, Block Scope, Closures, Closures Vs. Let - Timer Scope and

Closures vs. Let: Block Scope Differences)- ES6 Word Count Exercise (ES6 Word Count, ES6 Word Count Solution)- **Introduction to React:** ES6 Warm-Up Exercises(Introduction To ES6 Exercises, ES6-Warm-Up - Shopping Cart, ES6 Warm-Up - Shopping Cart Solution, ES6 Warm-Up Code Interpretation, ES6 Warm-Up Code Interpretation Solution)- Introduction To React(Introduction To The JSX Render Element and JSX Components, Passing Properties Into A Component)- Using ES6 In React(ES6 And React, Destructuring An Object and Renaming An Object, Map To Generate Child Buttons, Using Props Index To Identify Buttons, Using Closure To Generate ID Functions, Delete Button Exercise, Delete Button Solution)

UNIT – II **33 Hours**

Introduction to JSX: JavaScript Tips And Tricks(JavaScript Tips And Tricks And ECMAScript 262 - How The Interpreter Runs, How To Filter Lists And How To Use Reduce)-Introduction To JSX(Introduction To JSX, Introduction To Babel And Adding A JSX Component On A Page, Embedding Expressions And Adding Styles, Using Expressions And HTML Attributes In Components, Adding Children To Elements)-Introduction To Forms(Forms - Capturing Data, HTML Form And HTML Form Tag, React Form -Unmanaged And React Form – Managed)-Forms in React(Custom Hook To Clean Up React Form, Validate Form)- React Toolchains And Formik(What Is A Toolchain?, Toolchains, Formik Form And Formik Validation)-**Working With Lists in React:** Creating The ToDo List(Introduction To ToDo List And CRUD And ToDo List Overview, Set State And List ToDos And Styling The ToDo List, Input Form - Add ToDo Item And Remove ToDo Item)-Refactoring The ToDo List(Refactor - Form Component And Refactor - ToDo Component, Final Style Touches)-Rendering Lists(Introduction To Rendering List Exercises, List With Simple Data, Rendering List and Rendering List With Buttons And Handling onClick Events, Anonymous onClick Function To Pass Multiple Arguments, Saving State With UseState)-ReactDOM(ReactDOM, When Does UseEffect Run?)-Shopping Cart List(Shopping Cart List With Complex Data, Shopping Cart With Multiple Lists)

UNIT – III **33 Hours**

Working with Simple Components: ATM With Simple Components (Introduction To ATM Exercise, ATM With HTML And ATM Component, ATM In React And ATM State, Deposit And Cashback Buttons)-Fetching Data With React (Fetching Data Exercise Overview And Fetch Data

Via A URL With UseEffect, Render Data From URL)-Fetching Data From Hacker News (Fetch Data From Hacker News and UseReducer For Fetching Data, Paginate Data)- **Tic Tac Toe Exercise with React:** Introduction To Tic-Tac-Toe Exercise(Introduction To Tic-Tac-Toe Exercise and Tic-Tac-Toe Overview, Control Component With OnClick Events And Generate Players On The Tic-Tac-Toe Board)-Parent And Child Components(Add Squares As Child Components and Pass ID As State From Parent To Child,Change Color Of Square And Track State, Pass State From Child To Parent)-React Component Lifecycle(React Component Lifecycle: Mounting And Unmounting , Component Lifecycle, Parent State Forces Re-Render)-UseEffect And UseState Hooks(Track Re-Render With UseEffect, UseState Delay, Tracking Total Game State And Passing Player From Parent To Child, Tic-Tac-Toe Final Refactor.

UNIT – IV **33 Hours**

Working with React Components: Styling Components With Bootstrap(Introduction To Styling With Bootstrap And Styling Components)-Web Components And React Components(Introduction To Components - HTML To React And Web Components, Create Custom Element - Web Component And Create Web Components Programmatically, React Components And Web Components In React)-Routing And The Single Page Application(Introduction To Routing And The Single-Page Application, Overview of Frameworks And Routing And Route Using Buttons, Same Page Linking And Listening To Browser Events, URL Location Access With JavaScript And Route Using Location Hash, Building A Router, Same Page Linking Limitation And How To Avoid Reloading The Page)-**Bad Bank Exercise:** Growth Mindset- Introduction To Bad Bank(Introduction To Bad Bank Exercise)-Routing, Context, And Styles(Routing and Context Basics, Style Basics)-Bad Bank Walkthrough(Bad Bank Exercise Overview and Create Bad Bank Application Files, Bad Bank Navigation Bar and Routing, Bad Bank Context, Bad Bank - Styling The Navigation Bar and Styling The Bootstrap Card, Create Account,Bad Bank Challenge)

UNIT – V **33 Hours**

Shopping Cart Exercise: Integrating React With The Back End(Introduction To The Shopping Cart Exercise)-Strapi And Postman(Create Strapi Product Database, Install Postman, Using Postman And Express Web Server)-Shopping Cart Refactor(Shopping Cart Exercise - Refactor Introduction and Restocking Functionality, Shopping Cart Exercise - Refactor Solution)-**Testing:** Create React App(Introduction To Create React App, NPM Scripts For React , Install Create React

App, Create React App - Shopping Cart Example and Build Static Website)-Deploy To The Cloud(Introduction To Deployment, Introduction To Cloud Computing And Amazon Web Services , Deploy Static Website On AWS3)-APIs(Rest Principles And RESTful API, Introduction To GraphQL, What is GraphQL, GraphQL Demo)-Testing(Testing The Front End and Testing Overview, Install Node and Test-Driven Development, Testing Using The DOM and Testing Using The React Library, Testing Render, Testing Fire Event And Testing User Event, What Is A Mock And Why Does It Matter?, Mock A Dependency and Mock A ToDo Application API)

REFERENCES

1. Full Stack Development with MERN- MIT
2. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Template_literals
3. https://developers.google.com/web/updates/2015/01/ES6-Template-Strings
4. https://reactjs.org/docs/forms.html#fully-fledged-solutions
5. https://reactjs.org/tutorial/tutorial.html
6. https://adostes.medium.com/validating-a-form-in-react-cc29d47e140f
7. https://medium.com/geekculture/whats-a-crud-app-e5a29cce03b5
8. https://reactjs.org/docs/rendering-elements.html
9. https://reactjs.org/docs/hooks-intro.html
10. https://github.com/4GeeksAcademy/react-tutorial-exercises/tree/master/exercises
11. https://github.com/public-apis/public-apis
12. https://www.pluralsight.com/guides/how-to-handle-communication-between-parent-and-child-components-in-reactjs
13. https://towardsdatascience.com/passing-data-between-react-components-parent-children-siblings-a64f89e24ecf
14. https://towardsdatascience.com/passing-data-between-react-components-parent-children-siblings-a64f89e24ecf
15. https://medium.com/@NeotericEU/single-page-application-vs-multiple-page-application-2591588efe58
16. https://reactjs.org/docs/context.html
17. https://docs.strapi.io/developer-docs/latest/getting-started/introduction.html
18. https://aws.amazon.com/s3/

SEMESTER-II

MAJOR-1

Course Title: **Elective I- Front End Development with Angular JS**	Course Code : **23P**
Semester : **II**	Course Group : EL
Teaching Scheme in Hrs (L:T:P) : **3:2:6**	Credits : **8**
Map Code: **NA**	Total Contact Hours : **165**
CIA : **25 Marks**	SEE # : **75**
Programme: **MCA**	# - Semester End Exam

No.	Course Outcome	PSOs	Cl. Ses	CL
CO1	Learn how to use controllers for moving data to and from views	PSO1, PSO3	9L + 6T + 18P	AP
CO2	Understand when to use AngularJS services instead of controllers	PSO1, PSO3	9L + 6T + 18P	AP
CO3	Communicate with the server to store, fetch, and update data asynchronously	PSO1, PSO3	9L + 6T + 18P	AP
CO4	Know when to use AngularJS filters for converting data and values to different formats	PSO1, PSO3	9L + 6T + 18P	AP
CO5	Implement single-page applications, using ngRoute to select views and navigation	PSO1, PSO3	9L + 6T + 18P	AP

PEDAGOGY:

- Lecture, Group Discussion, LCD, Seminar & Case Study.

UNIT – I **33 Hours**

Introducing AngularJS: What Is MVC (Model-View-Controller)- AngularJS Benefits(AngularJS Benefits)-The AngularJS Philosophy-Starting Out with AngularJS(What Backend Do I Need?-Does My Entire Application Need to Be an AngularJS App) -A Basic AngularJS Application(AngularJS Hello World). **Basic AngularJS Directives and Controllers**: AngularJS Modules(Creating Our First Controller, Working with and Displaying Arrays,-More Directives)-Working with ng-repeat(ng-repeat Over an Object, Helper Variables in ng, repeat-Helper Variables in ng-repeat, Helper Variables in ng-repeat, Track by ID, ng-repeat Across Multiple HTML Elements.

UNIT –II **33 Hours**

Unit Testing in AngularJS: Unit Testing: (What and Why?)-Introduction to Karma(Karma Plugins, Explaining the Karma Config, Generating the Karma Config) - Jasmine: Spec Style of Testing(Jasmine Syntax, Useful Jasmine Matchers)-Writing a Unit Test for Our Controller(Running the Unit Test).**Forms, Inputs, and Services:** Working with ng-model(Leverage Data-Binding and Models, Form Validation and States)-Error Handling with Forms(Displaying Error Messages-Styling Forms and States),Nested Forms with ng-form(Nested Forms with ng-form, Other Form Controls, Textareas, Checkboxes, Radio Buttons, Combo Boxes/Drop-Downs).

UNIT – III **33 Hours**

All About AngularJS Services: AngularJS Services(Why Do We Need AngularJS Services?,-Services Versus Controllers-Dependency Injection in AngularJS, Using Built-In AngularJS Services, Order of Injection, Common AngularJS Services)- Creating Our Own AngularJS Service(Creating a Simple AngularJS Service ,The Difference Between Factory, Service, and Provider).**Server Communication Using $http**: Fetching Data with $http(Using GET ,A Deep Dive into Promises , Propagating Success and Error , The $q Service, Making POST Requests with $http , $http API, Configuration)- Advanced $http (Configuring $http Defaults , Interceptors , Best Practices).

UNIT – IV **33 Hours**

Working with Filters: What Are AngularJS Filters? (Using AngularJS Filters, Common AngularJS Filters, Using Filters in Controllers and Services)- Creating AngularJS Filters(Things to Remember About Filter).**Unit Testing Filters:** The filter under test(Testing the time ago filter). **End to end testing** : The need for protractor(initial setup, configurations, an end to end test, considerations)

UNIT – V **33 Hours**

Routing Using ngRoute : Routing in a Single-Page Application(Using ngRoute) - Routing Options(Using Resolves for Pre-Route Checks, Using the $routeParams Service,Things to Watch Out For,A Full AngularJS Routing Example) -Additional Configuration(HTML5 Mode-Analytics with AngularJS)-Alternatives(ui-router).**Directives:** What Are Directives-Alternatives to Custom Directives(ng-include, Limitations of ng-include , ng-switch)- Understanding the Basic Options (Creating a Directive, Template/Template URL , Restrict , The link Function, Scope , Replace).

REFERENCES:

1. AngularJS Up and Running- Enhanced productivity with structured Apps| OREILLY 2018| Shyam Seshadri and Brad Green

2. Learn Angularjs in 1 Day Complete Angular Js Guide with Examples| 2018 | Krishna Gupta

3. Learning Angular JS - A guide to Angular JS development | OREILLY 2015| Ken Williamson

4. Angular JS for Beginners Your Guide to Easily Learn Angular JS In 7 Days | Code Academy 2020

SEMESTER-II

MAJOR-1

Course Title: **Elective I- Front End Development with VUE JS**	Course Code : **23P**
Semester : **II**	Course Group : EL
Teaching Scheme in Hrs (L:T:P) : **3:2:6**	Credits : **8**
Map Code: **NA**	Total Contact Hours : **165**
CIA : **25 Marks**	SEE # : **75**
Programme: **MCA**	# - Semester End Exam

No.	**CourseOutcome**	**PSOs**	**Cl.Ses**	**CL**
CO1	Understand the basic concepts of NodeJS and NPM	PSO1, PSO3	9L + 6T + 18P	U
CO2	Create Web application with ExpressJS	PSO1, PSO3	9L + 6T + 18P	AP
CO3	Create API and JSON to the static web application	PSO1, PSO3	9L + 6T + 18P	AP
CO4	Understand and Implement EJS, Templates	PSO1, PSO3	9L + 6T + 18P	AP
CO5	Implement data manipulation using MongoDB and Implement Authentication using encryption ,Session and Cookies.	PSO1, PSO3	9L + 6T + 18P	AP

PEDAGOGY:

- Lecture, Group Discussion, LCD, Seminar & Case Study.

UNIT – I **33 Hours**

Vue.js: The Basics :

Why Vue.js? Installation and Setup (vue-loader and webpack) -Templates, Data, and Directives -v-if Versus v-show -Looping in Templates -Binding Arguments -Reactivity
(How It Works , Caveats)-Two-Way Data Binding.

Components in Vue.js: Component Basics - Data, Methods, and Computed Properties -Passing in Data (Prop Validation, Casing of Props , Reactivity, Data Flow and the .sync Modifier, Custom Inputs and v-model) -Passing Content into Components with Slots (Fallback Content, Named Slots , Scoped Slots) -Custom Events -Mixins(Merging Mixins and Components) -vue-loader and .vue Files -Non-prop Attributes -Components and v-for.

UNIT – II 33 Hours

Styling with Vue:

Class Binding - Inline Style Binding (Array Syntax, Multiple Values) - Scoped CSS with vue-loader - CSS Modules with vue-loader –Preprocessors. **Render Functions and JSX:**The Tag Name - The Data Object -Children- JSX

UNIT – III 33 Hours

Client-Side Routing with vue-router: Installation -Basic Usage -HTML5 History Mode - Dynamic Routing (Reacting to Route Updates, Passing Params to Components as Props)-Nested Routes -Redirect and Alias. Navigation (The output Tag, ActiveClass, NativeEvents,Programmatic Navigation) -Navigation Guards (Per-Route Guards, In-Component Guards) -Route Order -404 Pages -Route Names

UNIT – IV 33 Hours

State Management with Vuex: Installation - Concept - State and State Helpers(State Helpers) - Getters(Getter Helpers) - Mutations(Mutation Helpers, Mutations Must Be Synchronous) - Actions (Action Helpers, Destructuring) - Promises and Actions - Modules (File Structure, Namespaced Modules).

UNIT – V 33 Hours

Testing Vue Components: Testing a Simple Component - Introducing vue-test-utils- Querying the DOM -mount() Options - Mocking and Stubbing Data - Working with Events.

REFERENCES:

1. Vue.js: Up and Running by CallumMacrae, First Edition -2018, Published by O’Reilly Media, Inc.,

2. FullstackVue -The Complete Guide to Vue.js and Friendsby Hassan Djirdeh, Nate Murray, and Ari Lerner- 2018, Published by Fullstack.io

3. Vue.js 3By Example - John Au-Yeung (Author)-2021 - Packt Publishing

4. Getting to Know Vue.js: Learn to Build Single Page Applications in Vue from Scratch by Brett Nelson -2018 -Apress Publication

SEMESTER-II

MAJOR-2

Course Title: **BACK END DEVELOPMENT USING NODE JS AND MONGODB (T)**	Course Code : **23Q**
Semester : II	Course Group : **M**
Teaching Scheme in Hrs (L:T:P) : **4:2:0**	Credits : **6**
Map Code: **NA**	Total Contact Hours : **90**
CIA : **25 Marks**	SEE # : **75**
Programme: **MCA**	# - Semester End Exam

No.	**Course Outcome**	**PSOs**	**Cl.Ses**	**CL**
CO1	Illustrate the basic concepts of Node JS, NPM and Create Web application with Express JS	PO1,PSO2	12L+6T	AP
CO2	Implement API and JSON to the static web application	PO1,PSO2	12L+6T	AP
CO3	Demonstrate EJS Template in Node Application	PO1,PSO2	12L+6T	AP
CO4	Compute data manipulation using MongoDB	PO1,PSO2	12L+6T	AP
CO5	Implement Authentication using encryption, Session and Cookies.	PO1,PSO2	12L+6T	AP

PEDAGOGY:

- Lecture, Group Discussion, LCD, Seminar & Case Study.

UNIT-I **18 Hours**

Node JS: Back End Web Development-(Architecture, Examples), Introduction-(Installation, Command line usage), REPL-(Introduction, Server side Java Script), Native node modules-(Creation and Native node drivers), NPM Package manager-(Installation, Event loop and Emitters)

Express JS with Node JS: Introduction-(Creating our first server with express), Handling Request-(get and post), Routes-(Decorators, local host, RESTful routing), Handling Response-(HTML and server, Bodyparser, HTML file)

UNIT-II **18 Hours**

API: Introduction-(What is API?, paths, parameters and endpoints, API authentication and Postman), Request Module-(Get data from an API), JSON-(Format, working with JSON), API calls-(parameters and mailChimp API example, heroku deployment)

UNIT-III **18 Hours**

EJS: Templates-(Introduction, Creating EJS template), Passing DATA-(Example, Example, scope in the context),Style sheet and Template-(Adding CSS, Templating Vs Layouts), Node Module export-(pass function, data between files)

Databases and Sql: Sql vs NoSql-(CREATE Table and INSERT Data), Sql commands (READ, SELECT, and WHERE), Sql commands-(Updating Single, Values and Adding, Columns in SQL, Delete), Data Manipulation – (Foreign Keys and Inner Joins)

UNIT-IV **18 Hours**

MongoDB: Introduction - (Why Mongodb, Installation), CRUD - (Create, reading queries, Update and Delete, Relationships in MongoDB Normalization Vs Denormalization

Mongoose: Introduction – (Why mongoose, Installation), Features of Mongoose – (Reading from database, Data validation, Updating and deleting data, Establishing Relationships and Embedding Documents using Mongoose), Working with The Native MongoDB Driver- (Using the package in the app)

UNIT-V **18 Hours**

Authentication & Security: Introduction-(setup), Login-(register username and password), Data Encryption-(Introduction and Usage, environment variables), Hashing-(passwords and hacking, salting and hashing, Bcrypt), Cookies and Sessions- (JWT, manipulating cookie and session, Passport) Oath 2.0-(Implement Sign In with Google)

REFERENCES:

1. https://www.udemy.com/the-complete-web-development-bootcamp

2.AzatMardan, Full StackJavaScript, APress, 1st Edition, 2015.

3.Ethan Brown, Web Development with Node and Express: Leveraging the JavaScript Stack, O'Reilly, 1stEdition, 2015.

4. Mike Wilson, Building Node Applications with MongoDB and Backbone, O'Reilley Publications, 1st Edition, 2013.

5. Evan M. Hahn, Express in Action Writing, building, and testing Node.js applications, Manning Publications, 1st Edition, 2016.

SEMESTER-II

MAJOR - 3

Course Title: **PRACTICAL-II: SERVERSIDE PROGRAMMING USING NODE JS (P)**	Course Code : **23S**
Semester : **II**	Course Group : **MP**
Teaching Scheme in Hrs (L:T:P) : **0:0:6**	Credits : **3**
Map Code: **NA**	Total Contact Hours : **90**
CIA : **25 Marks**	SEE # : **75**
Programme: **MCA**	# - Semester End Exam

Course Outcomes

No.	Course Outcome	PSOs	Cl. Ses	CL
CO1	Illustrate the basic concepts of Node JS, NPM and Create Web application with Express JS	PO1,PSO2	18P	AP
CO2	Implement API and JSON to the static web application	PO1,PSO2	18P	AP
CO3	Demonstrate EJS Template in Node Application	PO1,PSO2	18P	AP
CO4	Compute data manipulation using MongoDB	PO1,PSO2	18P	AP
CO5	Implement Authentication using encryption, Session and Cookies.	PO1,PSO2	18P	AP

PEDAGOGY:

- Lecture, Group Discussion, LCD, Seminar & Case Study.

Unit - I **18 Hours**

Node JS: Back End Web Development-(Architecture, Examples)- **Express JS:** Server, HTTP, REST

Unit - II **18 Hours**

API: Introduction, usage, Creation, testing, and deployment)

Unit - III **18 Hours**

Templates: Create and Usage- **Databases :** CRUD, constraints

Unit - IV **18 Hours**

MongoDB : NOSQL vs RDBMS, CRUD and Constraints-**Mongoose:** Features, data validation, relationships

Unit - V **18 Hours**

Authentication: Login (userid, password), data encryption, Hashing, Salting-**Authorization:** JWT, passport, OAuth 2.0, Session and Cookies

REFERENCES:

1. https://www.udemy.com/the-complete-web-development-bootcamp

2.AzatMardan, Full Stack JavaScript, APress, 1st Edition, 2015.

3.Ethan Brown, Web Development with Node and Express: Leveraging the JavaScript Stack, O'Reilly, 1stEdition, 2015.

4.Mike Wilson, Building Node Applications with MongoDB and Backbone, O'Reilley Publications, 1st Edition, 2013.

5.Evan M. Hahn, Express in Action Writing, building, and testing Node.js applications, Manning Publications, 1st Edition, 2016.

SEMESTER-II

MAJOR – 4

Course Title: **Mini Project in Web development with MERN Stack**	Course Code : **23R**
Semester : **II**	Course Group : **PV**
Teaching Scheme in Hrs (L:T:P) : **0:5:6**	Credits : **8**
Map Code: **NA**	Total Contact Hours : **165**
FA: **25 Marks**	SEE # : **75 Marks**
Programme: **MCA**	# - Semester End Exam

No.	Course Outcome	PSOs	Cl.Ses	CL
CO1	Apply ReactJS in real time web application	PSO1& PSO3	15P	AP
CO2	Implement React route and styles in real time web components	PSO1& PSO3	15P	AP
CO3	Implement API in real time web Application for transferring JSON data with NodeJS	PSO1& PSO3	15P	AP
CO4	Apply Authentication & Security using NodeJS	PSO1& PSO3	15P	AP
CO5	Use Database Items to Application	PSO1& PSO3	15P	AP

PEDAGOGY:

- Lecture, Group Discussion, LCD, Seminar & Case Study.

UNIT-1 **33 Hours**

Home Page Ads-Design & Functionality Requirements (Functional components, class components, props and prop types). **Header and footer Component-**Design & Functionality Requirements (Create-react-app tool, CSS, Create API service and use fetch and get images from server.

UNIT-2 **33 Hours**

Profile page & edit profile Page- Design & Functionality Requirements (get profile details from API using fetch and display, Update API with new details changed by user).

UNIT-3 **33 Hours**

Gallery & product page - Design & Functionality Requirements (Create gallery item with reference to gallery item object,db.find(), db.update(), difference between findone() andfindbyid(), db.collection. find(),db.collection.update()).

UNIT-4 **33 Hours**

Product list with filters Page-Design & Functionality Requirements (Create product list item with respect to product JSON , limit() and skip() methods, Usearray.filter() for local filtering).**Cart Page-**Design & Functionality Requirements (Create cart item component,Usefetch()functionwithputinoptions,getcartAPIwithproducts).

UNIT-5 **33 Hours**

Orderslist, orderdetails, addressselection Pages-Design & Functionality Requirements (Create order item component , Use list component from react virtualized, get Orders API, create order details component, get order details API by order id, implement edit address API to update/delete address from server).

REFERENCES:

1.https://www.udemy.com/the-complete-web-development-bootcamp.

2.Zak Ruvalcaba and Anne Boehm, Murach's HTML5 and CSS3, Murach Publications, 4th Edition 2018.

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

4.https://nodejs.org/docs/latest-v15.x/api/

5.https://docs.mongodb.com/mongodb-vscode/

SEMESTER-II
Mandatory

Course Title : **Career and Professional Development II**	Course Code : **23E**
Semester : **II**	Course Group : **PG**
Teaching Scheme in Hrs (L:T:P) : **2 : 0:0**	Credits : -
Map Code : **O**	Total Contact Hours : **30**
CIA : **25 Marks**	SEE # : **75**
Programme: **MCA**	**# - Semester End Exam**

Course Outcomes (COs)

No.	**After completing this course, the students will be able to**	**PSOs**	**Cl. Ses**	**CL**
CO1	Enhance the Verbal Skills in the application of grammar in Placement Tests and Competitive Exam.	PO2	6	AP
CO2	Write a proper and appropriate response to an email or a letter	PO2	6	AP
CO3	Understand and relate their strengths, weaknesses, opportunities and threats. Based on the SWOT, build a strong Resume for a job interview.	PO3	2	U&C
CO4	Develop skills and ability to give a project presentation in front of a small audience	PO2	4	AP
CO5	Develop the ability to attend telephonic/virtual job interviews with confidence.	PO4	6	AP

UNIT: I Resume Writing + Building Linkedin Profile LECTURE HOURS:6

Resume Writing: (1) Resume - Your Ticket to Job InterviewSuccess **(2)** Creating Impressive Career Objective **(3)** Skills & Area of Interest **(4)** Talking about your Experience - Projects & Internship **(5)** Impressing with Co-Curricular & Extra-Curricular Activities **(6)** The Value of Hobbies in a Resume

Building LinkedIn Profile: Steps to create LinkedIn Profile

UNIT: II LECTURE HOURS:6

How to Write a Response Letter/Email: Replying to Letters/email: (1) Importance of letter/email writing **(2)** Review the Original letter/email **(3)** Constructing the response **(4)** Writing in a Professional Tone **(5)** Do's and Don'ts of Responsive Letter/Email **(6)** Letter/Email Writing Situation: Practice

UNIT: III LECTURE HOURS:6

Personal Skills: SWOT Analysis (Analysis of strength, weakness, opportunities and threats - Purpose of SWOT analysis.

Telephonic / Virtual Interview: (1) Getting the Documents Ready & Brushing up Your Knowledge **(2)** Knowing the Technology & Set Your Environment **(3)** Dress for the Occasion **(4)** Show Positive Body language **(5)** Preparing the Answers to FAQs of Job Interview **(6)** Perfecting Your Virtual Interview - Do's & Don'ts

UNIT: IV LECTURE HOURS: 6

Project Presentation Skills: **(1)** Understanding the project **(2)** Define the Presentation Objective **(3)** Steps for creating an effective presentation **(4)** Dress for the occasion **(5)** Effective Body Language **(6)** Do's and Don'ts of Project Presentation

UNIT: V: Verbal Ability LECTURE HOURS:6

Sentence Correction: Work out different questions**. Logical Sequencing**: Logical ordering of the sentence to form a meaningful paragraph **Verbal Analogy**: Method to solve, Verbal Analogy exercises **Word to Sentence Configuration:** Rules to frame sentences from a word jumble, Practice questions, **Synonyms - Antonyms**, **Prepositions**

FINAL EVALUATION: Mock Interview of students

TEXTBOOK: (To Change)

- Dr. Hyacinth Pink, “Verbal ability” Training Kit Version 3
- Dr. Hyacinth Pink, “Soft Skills” Training Kit Version 3
- Mrinalini Anand Arora, Essential English for Competitive Exams, Source books, 2016
- Robert M Sherfield, Rhonda J. Montgomery, Patricia G.Moody, Cornerstone Developing Soft Skills, fourth Edition, Pearson Education, 2010.
- Simon Sweeney, English for Business Communication, Cambridge University Press, 2012.

SEMESTER – III

MAJOR – 1

Course Title : **Full Stack Development Tools**	Course Code : 3**3P**
Semester : **III**	Course Group : **M**
Teaching Scheme in Hrs (L:T:P) : **3 : 2: 6**	Credits : 8
Map Code :	Total Contact Hours : **165**
CIA : **25 Marks**	SEE # : **75**
Programme: **MCA**	**# - Semester End Exam**

No.	After completing this course, the students will be able to	PSOs	Cl. Ses	CL
CO1	Build a secure full-stack web application using the MERN stack	PSO1, PSO4	9L + 6T + 18P	AP
CO2	Integrate React with a backend Express API	PSO1, PSO4	9L + 6T + 18P	AP
CO3	Explain the interactions between the different components of the MERN stack	PSO1, PSO4	9L + 6T + 18P	AP
CO4	Explain key cloud concepts related to modern web application development	PSO1, PSO4	9L + 6T + 18P	AP
CO5	Deploy a full stack solution using Continuous Integration (CI) and Continuous Delivery (CD) in DevOps pipelines	PSO1, PSO4	9L + 6T + 18P	AP

PEDAGOGY:

Lecture, Group Discussion, LCD, Seminar & Case Study.

Unit– I **33 Hours**

Introduction to Tiered Applications: Introduction to Back End (Getting Start with Back-End Development and What Is the Back End?) - Three-Tiered Application (Introduction to The Three-Tiered Application, Server, Data Store, Converting JavaScript Value Types, Server Plus Data Store, The Three Tiers and Receiving Data from The User Interface) - Unit Testing with Jest and Supertest (What Is a Unit Test, Jest Hello World Test, Jest SuperTest)

Three-Tiered-Applications: Server and Data Store (Working with The Three Tiers, Initialize Data Store, HTTP POST, Body Parser, Add Bootstrap Navigation and Serve Static Files) – Faker (Generate User Data with Faker, Read and Display Faker Data) - Cloud Computing Services (What Is Digital Ocean, Things to Consider When Selecting a Cloud Service Provider, What Is Heroku, Set Up a Virtual Machine withDigital Ocean, what is Cross Origin Resource Sharing (CORS), Move App to The Cloud)

Unit –II **33 Hours**

Introduction to Docker Containers: Introduction to Docker Containers (Introduction To Containers And Containers, Create, Build, And Run A Container, Docker Compose And Dockerize Applications, Dockerize Multi-Tier Applications, Dockerize Python-Flask-MongoDB Application, Docker Hub, Useful Docker Commands, Docker Resources, Run Docker Container In The Cloud) - Integrating Docker Containers With VS Code (Integrating VS Code With Docker Containers, VS Code Docker Support Extension, Use VS Code To Develop Within A Docker Container) - Integrating Docker Containers With VS Code (Containerize And Deploy An App).

Unit – III **33 Hours**

Express & GraphQL: Express Server (Introduction to Express Server, Serving Static Files) - Authentication And Authorization With JSON Web Tokens (Express And JSON Web Tokens, What Are JSON Web Tokens And Why Use Them?, Express And JSON Web Tokens Overview, Introduction to JSON Web Token Authentication Exercise) - Authentication And Authorization With OAuth2 (What Is OAuth2?, Introduction To Authentication And Authorization and Okta And OAuth2) – GraphQL (Introduction To GraphQL And Strapi Exercise, Create Strapi Database, GraphQL Set Up and GraphQL And Strapi Exercise, Strapi And GraphQL Mutations, Create New

Contact With GraphQL, GraphQL Tool Update-Delete Exercise, GraphQL Tool Update-Delete Exercise Solution)

Unit – IV **33 Hours**

API Documentation And Introduction To Database: Automated API Documentation (Introduction To Swagger, Swagger Overview, Install Packages and AutomaticDocumentation, API Documentation Best Practices, Post Route, Swagger API Restaurant Example) - Introduction To Database (Introduction To The Holy Grail With Database and Holy Grail Overview,Holy Grail Design, Holy Grail Styles, Holy Grail Components and Holy Grail Icons, Holy Grail Front End, Holy Grail Server Overview and Holy Grail Server Setup, Serve Static Files, Promises And The JavaScript Event Loop, Asynchronous Front-End Calls, Redis Basics and Holy Grail Application)

Unit– V **33 Hours**

Serverless Computing: Firebase (Introduction to Serverless Services and Overview of Serverless Architecture, Firebase Project and Firebase App Pattern, Real-time Database, Storage, Authentication, Cloud Functions, Authentication of Server Routes, Free Resources, Cloud Services Similar Resources) - Asynchronous Programming (Asynchronous Calls, Client Async Calls, Server Async Calls)

Testing and DevOps: Testing (Introduction To Back-End Testing, Testing Overview, Testing Pyramid, Back-End Testing, Back-End Test, User Interface Testing, Testing Wrap Up) - Continuous Integration And Deployment (Introduction To Continuous Integration And Continuous Deployment With GitHub Actions , Continuous Integration And Continuous Deployment, Web Page Deploy to AWS-S3, Heroku For CI/CD, Continuous Integration - Express Docker Image To DockerHub, Continuous Deployment - Deploy GitHub Image On AWS Elastic Cloud)

REFERENCES

1. Full Stack Development with MERN – Back End Development - MIT
2. Modern Full-Stack Development | 2020 | Frank Zammetti | Apress
3. Full Stack Serverless | July 2020 | Nader Debit | O'Reilly Media, Inc
4. Full stack Development with MongoDB | April 2022| Manu Sharma | BPB Publications

SEMESTER – III

MAJOR - 2

Course Title : **Core Elective II-Mobile App Development using React Native (T)**	Course Code : **33Q**
Semester : **III**	Course Group : **M**
Teaching Scheme in Hrs (L:T:P) : **4:2:0**	Credits : **6**
Map Code:	Total Contact Hours : **90**
CIA : **25 Marks**	SEE # : **75**
Programme: **MCA**	**# - Semester End Exam**

No.	**Course Outcome**	**PSOs**	**Cl. Ses**	**CL**
CO1	Illustrate Props and State in React to develop an App and Use React Native to develop hybrid App	PSO1, PSO4	12L + 6T	AP
CO2	Use App with lists, user inputs and debugging	PSO1, PSO4	12L + 6T	AP
CO3	Demonstrate Navigation, Data and Map in App development	PSO1, PSO4	12L + 6T	AP
CO4	Use Redux, Async Redux and Tools for managing state of App	PSO1, PSO4	12L + 6T	AP
CO5	Apply Performance to App , deploy and testing App	PSO1, PSO4	12L + 6T	AP

PEDAGOGY:

- Lecture, Group Discussion, LCD, Seminar & Case Study.

UNIT - I **18 Hours**

React, Props, State: Classes (Classes vs instances, Methods vs static methods vs properties, new, constructor, extends, super) - React, Imperative vs Declarative(Examples) - Writing React (JSX,Components are just functions) – Props (Passed as an object to a component and used to compute the returned node) – State (Adds internally-managed configuration for a component). **React Native :** How does React Native work?(JavaScript is bundled, Separate threads for UI, layout and JavaScript) - Differences between RN and Web (Base components, Style) - Event

Handling and Components (React Native handlers often receive different arguments) - Stateless Functional Component (SFC) (Simplest component: use when you don't need state) - React.Component and Component Lifecycle (Have instances, Maintain their own state, Have lifecycle methods that are automatically invoked) - Mount, Update, and Unmount (methods) - **Writing React Native:** Expo (The fastest way to build an app) – Import/Export (Components are great for simplifying code) – PropTypes (React can validate the types of component props at runtime) - How to Read Docs (required steps)

UNIT – II **18 Hours**

Lists, User Input : Lists (ScrollView, ListView (deprecated),FlatList, SectionList) - User Input (Controlled vs uncontrolled components) - User Input, Debugging: Handling multiple inputs(<form> exists in HTML, but not in React Native) - Validating Input(Conditionally set state based on input value, Validate form before submitting, Validate form after changing single input value) - KeyboardAvoidingView(Native component to handle avoiding the virtual keyboard, Good for simple/short forms) – Debugging (React errors/warnings, Chrome Developer Tools (devtools), React Native Inspector, react-devtools) - External Libraries(Install using npm install <library>,Import into your project)

UNIT – III **18 Hours**

Navigation Using the Navigator: React Navigation 5.0 – Installing react navigation –The react native stack navigator – React Native Navigation Examples (Using stack navigator to navigate between the screen components, Tab Navigation, Using Drawer Navigation) – **Data:** Introduction (Examples) - API(A defined set of ways with which a resource can be interacted) - Making Network Requests (fetch()) - Promises (writing asynchronous, non-blocking code,chaining callbacks and/or error handlers) - Async/Await(writing async code as if it were synchronous) - Transforming Data and Authentication(What is transformation? And What is Authentication?) - HTTP Methods and Response Codes (get() and post(), codes) - **Maps and Location (Expo Components) :** Introduction (MapView, Location, Geocoding) - Text or Delete(Contacts) - A Compass (Magnetometer,DeviceMotion) - A Multimedia Board (Video,Audio, Fonts) - A Photo Editor (ImagePicker,Simple Camera,Camera View, GestureHandler).

UNIT – IV **18 Hours**

Redux: Other APIs (Push Notifications,Calendar, Gyroscope, Accelerometer, Pedometer
) - Scaling Complexity (Our apps have been relatively simple, but we're already starting to see bugs related to app complexity) – Flux (An application architecture for React utilizing a unidirectional data flow) – Redux:Introduction (A data management library inspired by Flux, Single source of truth for data) – Reducer (Takes the previous state and an update and applies the update, Should be a pure function) – Store (Responsible for maintaining state, Exposes getter via getState()) - Actions (An action is a piece of data that contains the information required to make a state update) - react-redux (React bindings for redux, Provider gives children access to our redux store) - **Async Redux, Tools:** Supporting Async Requests (Where do we want to add this support? How do we change our API?) - Redux Middleware (This allows us to extend redux without having to touch the implementation, Any function with this prototype can be middleware) - Persisting State (redux-persist
) - Container vs Presentational Components (As an application grows in size and complexity, not all components need to be aware of application state) - Do I need Redux? (Redux helps apps scale, but does add complexity, Sometimes, the complexity overhead isn't worth it) - JavaScript Tools(ESLint and Prettier)

UNIT – V **18 Hours**

Performance: Introduction (How quickly and efficiently something works,Performance optimization is the process of making something work as efficiently as possible) – Trade-Offs (Performance optimization usually comes at a complexity cost) - Measuring Performance (Be mindful of the environment setting of your application, React Native Perf Monitor,Chrome Performance Profiler) - Common Inefficiencies (Rerendering too often, Unnecessarily changing props, Unnecessary logic in mount/update) – Animations (Animations require both the JS and UI threads) - **Deploying, Testing :** Deploying (Deploy to the appropriate store by building the app and uploading it to the store, Set the correct metadata in app.json,Build the app using exp,Upload to the appropriate store,Deploy new JS by republishing from the XDE or exp) – Testing (Test Pyramid,Unit Tests) - Test Pyramid (Methodology for determining test scale/granularity, Unit tests, Integration/Service tests, UI/End-to-end tests) – Jest (Testing Redux Actions,Testing Async

Redux Actions) - Code Coverage (Metric for tracking how well-tested an application, Get coverage report by passing --coverage)

REFERENCES:

1. https://courses.edx.org/courses/course-v1:HarvardX+CS50M+Mobile/course/
2. https://www.udemy.com/react-native-the-practical-guide/
3. https://www.udemy.com/react-native-advanced/

4. "Mastering React Native- A Beginners Guide" by Sufyan bin Uzayr CRC Press (Taylor & Francis Group) First Edition, 2023

SEMESTER – III

MAJOR – 2

Course Title : **Core Elective II-Mobile App Development using Flutter (T)**	Course Code : **33Q**
Semester : **III**	Course Group : **M**
Teaching Scheme in Hrs (L:T:P) : **4:2:0**	Credits : **6**
Map Code:	Total Contact Hours : **90**
CIA : **25 Marks**	SEE : **75 Marks**
Programme: **MCA**	**# - Semester End Exam**

No.	Course Outcome	PSOs	Cl. Ses	CL
CO1	Understand the basic concepts of Flutter andDart	PSO1, PSO4	12L + 6T	AP
CO2	Create Mobile application with templates and Widget Trees using Dart	PSO1, PSO4	12L + 6T	AP
CO3	Understand Common Widgets and Implement Animation and Navigation	PSO1, PSO4	12L + 6T	AP
CO4	Build Lists, layouts and apply Interactivity and Write platform native code	PSO1, PSO4	12L + 6T	AP
CO5	Store Data with Local Persistence	PSO1, PSO4	12L + 6T	AP

PEDAGOGY:

- Lecture, Group Discussion, LCD, Seminar & Case Study.

UNIT-I **18 Hours**

Introducing Flutter and Getting Started: Introducing Flutter - (Defining Widgets and Elements), Understanding Widget Lifecycle Events – (The StatelessWidget Lifecycle, The StatefulWidget Lifecycle,Understanding the Widget Tree and the Element Tree –(Stateless Widget and Element Trees, Stateful Widget and Element Trees), Installing the Flutter SDK-(Installing on Windows, System Requirements, Get the Flutter SDK, Check for Dependencies, Install Android Studio,Set Up the Android Emulator)

Creating a Hello World App: Setting Up the Project – (Using Hot Reload), Using Themes to Style Your App – (Using a Global App Theme, Using a Theme for Part of an App), Understanding Stateless and Stateful Widgets- (Example), Using External Packages –(Searching for Packages, Using Packages)

UNIT-II **18 Hours**

Learning Dart Basics: Why Use Dart?-(Commenting Code, Declaring Variables, Numbers, Strings, Booleans, Lists, Maps, Runes), Using Operators – (Example), Using Flow Statements – (if and else, ternary operator, for Loops, while and do-while while and break, continue, switch and case, Using Functions, Import Packages –(Example), Using Classes – (Class Inheritance, Class Mixins)

Creating a Starter Project Template: Creating and Organizing Folders and Files -(Examples), Structuring Widgets–(Examples)

Understanding The Widget Tree: Introduction to Widgets –(Example), Building the Full Widget Tree - (Example), Building a Shallow Widget Tree – (Refactoring with a Constant, Refactoring with a Method, Refactoring with a Widget Class)

UNIT III **18 Hours**

Fleshing Out an App-using common widgets: Using Basic Widgets –(Safe Area, Container, Text, Rich Text, Column, Row, Column and Row Nesting), Buttons-(Floating Action Button, Flat Button, Raised Button, Icon Button, Popup Menu Button, Button Bar), Using Images and Icons-(Asset Bundle, Image, Icon, Using Decorators), Using the Form Widget to Validate Text Fields – (Example), Checking Orientation –(Example)

Adding Animation to an App: Using AnimatedContainer –(Example), Using AnimatedCrossFade- (Example), Using AnimatedOpacity- (Example) Using AnimationController-(Using Staggered Animations) **Creating an App's Navigation:** Using the Navigator – (Example),Using the Named Navigator Route - (Example), Using Hero Animation - (Example), Using the BottomNavigationBar - (Example), Using the BottomAppBar - (Example), Using the TabBar and TabBarView - (Example), Using the Drawer and ListView - (Example)

UNIT IV **18 Hours**

Creating Scrolling Lists and Effects: Using the Card - (Example), Using the ListView and ListTile - (Example), Using the GridView –(Using the GridView.count, Using the GridView.extent, Using the GridView.builder), Using the Stack - (Example), Customizing the CustomScrollView with Slivers - (Example) **Building Layouts:** A High-Level View of the Layout – (Weather Section Layout, Tags Layout, Footer Images Layout, Final Layout), Creating the Layout- (Example) **Applying Interactivity:** Setting Up GestureDetector: The Basics - (Example), Implementing the Draggable and Dragtarget Widgets - (Example), Using the GestureDetector for Moving and Scaling - (Example), Using the InkWell and InkResponse Gestures - (Example), Using the Dismissible Widget - (Example) **Writing Platform-Native Code:** Understanding Platform Channels – (Example), Implementing the Client Platform Channel App – (Example), Implementing the iOS Host Platform Channel – (Example), Implementing the Android Host Platform Channel – (Example)

UNIT V **18 Hours**

Saving Data with Local Persistence: Understanding the JSON Format – (Example), Using Database Classes to Write, Read, and Serialize JSON – (Example), Formatting Dates – (Example), Sorting a List of Dates – (Example), Retrieving Data with the FutureBuilder – (Example), Building the Journal App – (Example), Adding the Journal Database Classes – (Example), Adding the Journal Entry Page – (Example), Finishing the Journal Home Page – (Example).

REFERENCES:

1. Marco L. Napoli, Flutter – A hands on guide to App development, John Wiley & Sons, Inc., 1st Edition, 2020.
2. Rap Payne, Beginning App development with Flutter, Apress publications, 1st Edition, 2019.

SEMESTER – III

MAJOR – 2

Course Title : **CORE ELECTIVE – II - MOBILE APP DEVELOPMENT USING ANDROID (T)**	Course Code : **33Q**
Semester : **III**	Course Group : **M**
Teaching Scheme in Hrs (L:T:P) : **4:2:0**	Credits : **6**
Map Code:	Total Contact Hours : **90**
CIA : **25 Marks**	SEE: **75 Marks**
Programme: **MCA**	**# - Semester End Exam**

No.	Course Outcome	PSOs	Cl. Ses	CL
CO1	Understand Android Programming Concepts and techniques	PSO1, PSO4	12L + 6P	AP
CO2	Implement User Interfaces, Testing and Debugging on existing App	PSO1, PSO4	12L + 6P	AP
CO3	Apply Layout, widgets, menus and preferences in App	PSO1, PSO4	12L + 6P	AP
CO4	Use Fragment, single pane and multi pane layouts in App	PSO1, PSO4	12L + 6P	AP
CO5	Apply database, tabs and custom adapter in App and Create production ready Apps	PSO1, PSO4	12L + 6P	AP

PEDAGOGY:

- Lecture, Group Discussion, LCD, Seminar & Case Study.

UNIT : I **18 Hours**

Android overview: Types of devices (Readers, cameras, Home automation systems) - Types of Apps (Photography, Audio, video, Weather)- A brief history (Open Handset Alliance, Android open Source Project(AOSP))- Versions (Donut, Honeycomb)-System Architecture (Dalvik virtual machine(DVM), Java virtual machine(JVM))- How Apps are compiled and run (Integrated Development Environment(IDE), Android Virtual Device(AVD), Android Debug Bridge(ADB)).

A simple Android app: The user interface (Soft keyboard)- The XML for the user interface (Parent element, Child element, editable text view, scale – independent pixel)-The XML for the display text (Localization, activity)-The java source code (Android.widget.packages , setContentView method)-The Android Manifest(Androidmanifest, portrait).

How to work with existing projects: An introduction to eclipse projects (Work bench, code editor, views, projects)-How to set the work space (Workspace, Switch Workspace menu)-How to import the project in to the workspace (book_apps directory, unable to resolve target)-How to remove the project in to the workspace (Delete project contents, Delete Launch Configurations)-How to work with the user interface (Graphical layout editor, res\layout directory, properties view, XML editor)- How to work with other XML resources (strings.xma , Styles.xml)-How to work with the java code(Eclipse text editor , src directory)-How to set the run configuration(run configuration , Always prompt option)-How to run an app on a physical device(Package Explorer , Eclipse)-How to run an app on an emulator(Emulator , Launch option , Clean item).

How to work with a new project: How to create a new project (Tester, Create Custom Launcher Icon)-How to work with the starting code (Relative Layout Element, Text View element, TestActivity class)-How to use the code completion feature (Code completion feature, Content Assist feature)-How to detect and correct errors and warnings (Quick Fix feature, Errors, warnings).

The tester app: The user interface (The tester app, emulator)-The XML for the user interface (Text View widget, text size, text style)-The java source code (The activity_test.xml file, The Activity.java file).

UNIT : II **18 Hours**

How to develop the user interface: The tip calculator APP (Widgets, controls, text view, label button)-How to work with a layout (Layout, form, Graphical layout tab)-How to add widgets to a layout (Widgets, pallet view, theme)-How to set the display text (Graphical Layout Editor, Android string dialog box)-How to work with the strings.xml (file Manifest file, options menu, values xx directory)-How to set properties(Properties view , Text Style property)-Common properties(Common properties for layouts , Common properties for widgets, predefined values for setting height and width , common units of measurement)-The XML for the user interface(Padding attribute , match_pattern).

Basic skills for testing and debugging:

Typical test phases (Testing, debugging EditTextWidget)-How to check the layout (Graphical Layout Editor, Emulate Real Size button, Flip orientation button)-The three types of errors (Syntax errors, runtime errors, exceptions, logic errors)-How to handle run time errors (The error that's displayed when an app crashes).

How to trace code execution: How to use LogCat logging (LogCat view, import the Log class, declare a constant for the tag parameter, send messages to Log)-How to use toasts (LENGTH_SHORT, LENGTH_LONG, method chaining, make Text, show).

How to use the debugger: How to set and remove breakpoints (Breakpoint, java perspective, Debug perspective)-How to step through code (Step through, Step into button)-How to inspect variables (This, scope)-How to inspect the stack trace (Stack trace, onEditorAction method, calculateAndDisplay method)-How to configure step filters(Java and Android libraries , Step Filtering , android.* , com.android.* , com.google.*).

How to configure your emulator: How to add an emulator for an old phone (Built in device definitions, DPad or trackball)-How to work with an emulator for an old phone (Hardware buttons are enabled, Dpad is enabled, hardware keyboard is enabled)-How to add an emulator for a tablet (ARM processor, Android 3.0(API 11)).

UNIT : III **18 Hours**

An introduction to layouts and widgets: A summary of layouts(RelativeLayout , LinearLayout , Table Layout , FrameLayout , AbsoluteLayout , GridLayout)-A summary of widgets(TextView , EditText , Button ,CheckBox)-The view hierarchy(viewGroup , TextView , ProgressBar , ImageView).

How to work with layouts: How to use a linear layout(Match_parent , vertical , the xml for the linear layout, common attributes for working with linear layouts)-How to use a table layout(common attributes for working with table layouts)-How to use frame layout(TextView widget , the xml for the frame layout)-How to use nest layout(The xml for nested linear layouts)-

How to provide a landscape layout (The location of the xml files ,the xml for landscape orientation).

How to work with widgets:

How to use editable text views (The soft keyboard for an editable text view for an email address)-How to use check boxes (A common xml attribute for check boxes, two common java methods for check boxes)-How to use radio buttons (Three radio buttons in a radio group with vertical orientation, Three radio buttons in a radio group with horizontal orientation)-How to use spinners (The xml code, the array in the strings.xml file)-How to use seek bars (Thumb, two common java methods for seek bars)-How to display images (ContentDescription , xhdpi , hdpi , mdpi , ldpi)-How to show and hide widgets(setVisibility)-How to add scroll bars(ScrollView widget , vertical scroll bar , horizontal scroll bar).

How to work with menus: An introduction to menus (Menu, items, options menu) - How to define a menu (Title, icon, showAsAction ,orderIncategory)-How to display options menu(CreateOptionsMenu method , inflate)-How to handle option menu events (OptionsItemSelected method , the code that displays the menu)-How to start a new activity(Intent , constructor , intent class).

How to work with preferences: An introduction to preferences (Preferences or settings, preference APIs)-How to define preferences (Key, title, summary, defaultValue)-How to display preferences in an activity (@SuppressWarnings, @Override)-How to display preferences (in a fragment Fragment, multiple activities, FragmentManager)- How to get preferences (OnCreate , SharedPreferences object , onResume)-How to use references(Using if else statement , Rounding preference).

More skills for working with preferences:

How to group preferences (Nesting one or more preference elements within a preferenceCategory element) - How to enable and disable preferences (Dependency preference, ChechBox preference)-How to use java to work with preferences (A class that works with preferences).

UNIT : IV **18 Hours**

An introduction to fragments:

Single pane and multi pane layouts (Single pane layout , Multi pane layout)-How to use support libraries(Libs\android-support-v4.jar)-The life cycle methods of a fragment(OnResume() , onStart () , onActivityCreate(),onCreateView(),onCretae(),onAttach(),onPause(),onStop()).

How to use single pane layouts for small screens:

How to create the layout of a fragment (The layout for a fragment, xml, Eclipse)-How to create the class of a fragment (setHasOptionsMenu(bool), Fragment class, findViewById)-

How to display a fragment in an activity (The activity_main.xml file)-How to create a preference fragment (Preference objects, SettingsFragment)-How to display a preference fragment in an activity(The SettingsFragment class,the activity_setting.xml file , the settingsActivity class).

How to use multipane layouts for large screens:

How to add multiple fragments to a layout(Landscape mode, portrait mode)-How to detect large screens(Res\values-large-land , large , xlarge)-How to detect screen width(Sw480 dp, sw600 dp, sw720p)- How to control the soft keyboard(actionDone , actionNext , actionprevious , actionGo,actionSearch)-How to get reference to a fragment(getFragmentManager , findFragmentById(id))-How to replace one fragment with other (beginTransaction , replace(id,fragment), commit()).

UNIT : V **18 Hours**

An introduction to databases:

The user interface for the Task List APP (TaskList app, the tasklist activity, the Add/Edit activity)-An introduction to SQLite (SQLite, Relational Database Management System(RDBMS), Embedded, Lightweight, secure)- An introduction to the Task List database (List_name Create table, primary key, auto_increment)-The business objects for the Task List app (Business objects,plain Old Java Objects(POJO)).

How to add public methods to a database class:

How to define the constants for a database(TaskListDB class , LIST_TABLE , LIST_ID ,LIST_ID_COL)-How to define the SQL statements that create a database(LIST_ID , LIST_NAME ,LIST_TABLE ,LIST_NAME)-How to create or upgrade a database(DBHelper class LogCat view ,execSQL(sqlString)-How to open and close a database connection (getReadableDatabase(),getWritableDatabase() , close())-How to retrieve multiple rows from a table(Where clause ,getId method , order by clause)-How to retrieve a single row from a table(TaskFromCursor , where clause)-How to get data from a cursor (getInt(ColumnIndex) , getDouble(ColumnIndex) , getString(ColumnIndex))-How to insert , update and delete rows(Insert , update , put , delete).

How to add test the database class and clear its data:

How to test the database class (TaskListDB, updateCount, task ID)-How to clear test data from a device (Version number for a database, DBHelper class)-How to use the DDMS perspective to work with database files (DDMS, File Explorer, TaskList app)-How to use the SQLite database browser (insert statement, SQL Database Browser).

How to use tabs: How to add the TabManager class to your project (TabManager , Android API)-The layout for an activity that displays tabs(TabHost widget , TabManager.jar file)-The class for an activity that displays tabs(The context , the TAbHost object , ID of the FrameLayout)-The class for a fragment that displays tab content(TextView widget , TabHost object , Personal tab).

How to use a custom adapter:

A layout for a list view item (CheckBox widget, TextView widgets, textColor attributes)-
A class that extends the layout for a list view item (RelativeLayout, TaskListDB, getSystemServiceMethod)-A class for a custom adapter (BaseAdapter class, getCount method, The getItem method, getItemId method)-A class for a fragment that uses a custom adapter(TaskListAdaptor, ListView control, getTasks method).

The TaskList app: The user interface-The activity_task_list menu (TaskList activity, Delete Tasks icon, The Add/Edit activity)-The Task List Activity class (Add Task item, Add/Edit activity)-The activity_ add_edit and spinner_list layout (Source code)-The activity_ add_edit menu (The activity_add_edit menu)- The AddEdit Activity class (onKey method, D-pad, ToDB).

REFERENCES:

1. Training and Reference Murach's Android programming by Joel Murach , 4th Edition, 2014 Mike Murach&Associates ,Inc. (Chapters 1,2,3,4,5,8,9,13,14)

2. https://developer.android.com/guide

SEMESTER – III

MAJOR – 3

Course Title : **PRACTICAL – III MOBILE APP DEVELOPMENT (P)**	Course Code : **33S**
Semester : **III**	Course Group : **M**
Teaching Scheme in Hrs (L:T:P) : **0:0:6**	Credits : **3**
Map Code:	Total Contact Hours : **90**
CIA : **25 Marks**	SEE: **75 Marks**
Programme: **MCA # - Semester End Exam**	

No.	**Course Outcome**	**PSOs**	**Cl. Ses**	**CL**
CO1	Illustrate Props and State in React to develop an App and Use React Native to develop hybrid App	PSO1, PSO4	18P	AP
CO2	Use App with lists, user inputs and debugging	PSO1, PSO4	18P	AP
CO3	Demonstrate Navigation, Data and Map in App development	PSO1, PSO4	18P	AP
CO4	Use Redux, Async Redux and Tools for managing state of App	PSO1, PSO4	18P	AP
CO5	Apply Performance to App , deploy and testing App	PSO1, PSO4	18P	AP

PEDAGOGY:

- Lecture, Group Discussion, LCD, Seminar & Case Study

UNIT - I 18 Hours

Understanding the Requirements: Applying Business Rules-(Requirements gathering)

UNIT - II 18 Hours

Design: Page design –(User Interface)

UNIT - III 18 Hours

SRS-based Operations: Component and Content creation- Events, themes, and styles, menus and navigation)

UNIT - IV 18 Hours

Development: Configure, Connection and Creation-(Connector, API), Data Manipulation and Validation-(Test cases)

UNIT - V 18 Hours

Implementation: Execution-(Hosting, Deployment)

REFERENCES:

1. https://courses.edx.org/courses/course-v1:HarvardX+CS50M+Mobile/course/
2. https://www.udemy.com/react-native-the-practical-guide/
3. https://www.udemy.com/react-native-advanced/
4. "Mastering React Native- A Beginners Guide" by Sufyan bin Uzayr CRC Press (Taylor & Francis Group) First Edition, 2023

SEMESTER – III

Major -4

Course Title **: Mini Project in Mobile App Development using React Native**	Course Code :**33R**
Semester: **III**	Course Group : **PV**
Teaching Scheme in Hrs (L:T:P) : **0:5:6**	Credits : **8 Credits**
Map Code: NA	Total Contact Hours: **165**
CIA :**25 Marks**	SEE : **75Marks**
Programme:MCA # - Semester EndExam	

No.	Course Outcome	PSOs	Cl. Ses	CL
CO1	Use essential tools for React Native development and Work with the core components and APIs	PSO1, PSO4	15T+18P	AP
CO2	Demonstrate layouts with Flexbox and reusable components	PSO1, PSO4	15T+18P	AP
CO3	Demonstrate forms with Formik Implement data validation with Yup and Access native device features and Implement navigation using React Navigation	PSO1, PSO4	15T+18P	AP
CO4	Compute with REST APIs and Build offline capable apps	PSO1, PSO4	15T+18P	AP
CO5	Implement authentication and authorization and Send and receive push notifications	PSO1, PSO4	15T+18P	AP

PEDAGOGY:

- Lecture, Group Discussion, LCD, Seminar & Case Study

UNIT – I **33 Hours**

Introduction: Prerequisites, What is React Native, Mobile app development with React Native-(setting up the development Environment, Expo Creating, running and deploying the app, logging and publishing). **Fundamental concepts:** Core components and API of React native, core components – View, Text, Image, Image, Touchables, Button, Alert and stylesheet - (Creating StyleSheet and writing platform specific code) **Layouts**: Dimensions, Aligning content and positioning, Flexbox, Detecting Orientation changes.- (Building the Welcome Screen, Building the View Image Screen

UNIT – II **33 Hours**

Styling: Borders, shadows, padding, margins, styling Text, icons, Platform specific code, Encapsulating styles. – (Building the Button Component, Improving the Welcome Screen, Building the Card Component, Building the Listing Details Screen, Building the ListItem Component, Improving the View Image Screen)

Lists: Flat List operation, handling swipes, handling selection, Deleting an Item, Separators, implementation of Pull to refresh. – (Planning the Account Screen, Building the Icon Component, Extending the ListItem Component, Building the Account Screen, Building the Listings Screen)

Input components: Learn and implement TextInput, Switch, Picker, Modal, Handling Selections – (Building a Custom Picker)

UNIT – III **33 Hours**

Forms - Forms, better forms with formik, validation with Yup, Touched state. – (Building the ,Login Form, Building Better Forms with Formik, Form Validation with Yup, Building the ErrorMessage Component, Building the Field Component, Building the SubmitButton Component, Building the Form Component, Organizing Components, Building the Listing Edit Screen)

Improvements: Learn and work with Adding chevrons, handling long text, Input components with dynamic width – (Improving the Messages Screen, Building the CategoryPickerItem Component)

Native features: Device features, Image Picker, Requesting permissions, permission module and accessing the library – (Building ImageInput- Layout, Building ImageInput- Touches, Building ImageInputList- Basics, Building ImageInputList- Scrolling, Building Form ImagePicker, Improving ListingEditScreen, Building Custom Hooks)

Navigation - React navigation, Stack navigator, Passing parameters to Routes, Setting the screen titles, customizing the headers, Creating Tab navigator, Beautifying the tabs, refactoring the routes – (Creating a TabNavigator, Building AuthNavigator, Building AppNavigator, Building FeedNavigator, Building AccountNavigator)

UNIT – IV **33 Hours**

Networking - Setting backend, calling API with APIsauce, Creating an API layer, fetching data, Inspecting API calls, Handling error, Simulating a slow connection, Showing an activity indicator – (Building a Beautiful Activity Indicator, Creating a Reusable Hook, Extending the API Layer, Posting Data, Tracking Upload Progress, Building the Upload Screen, Resetting the Form)

Offline support - Strategies for Building Offline Capable Apps, Determining network status, caching, Async storage, storing user actions when offline – (Showing an Offline Notice)

UNIT – V **33 Hours**

Authentication and Authorization - Authentication providers, Authentication flow, getting, extracting auth token, Storing, getting the current user, calling protected apis, Persisting the Authentication State Across Restarts. – (Implementing the Registration, Showing an Activity Indicator) **Notification -** Push notification service, architecture, storing, sending a push notification, sending and receiving test notification, Handling Received Notifications, Navigation Upon Receiving a Notification, Local Notifications.- (Building the Contact Form)

REFERENCES:

1. https://courses.edx.org/courses/course-v1:HarvardX+CS50M+Mobile/course/
2. https://www.udemy.com/react-native-the-practical-guide/
3. https://www.udemy.com/react-native-advanced/
4. "Mastering React Native- A Beginners Guide" by Sufyan bin Uzayr CRC Press (Taylor & Francis Group) First Edition, 2023

SEMESTER –III

Mandatory

Course Title : **Career and Professional Development III**	Course Code : **33E**
Semester : **III**	Course Group : **PG**
Teaching Scheme in Hrs (L: T: P): **2:0:0**	Credits : 1
Map Code : O	Total Contact Hours : **30**
CIA : **25 Marks**	SEE #: **75 marks**
Programme : II **MCA**	**# - Semester End Exam**

Course Outcomes (COs)

No.	After completion of this course, the students will be able to	PSOs	Cl. Ses	CL
CO1	Understand the basic concepts of quantitative ability	PO5	6	Apply
CO2	Understand the basic concepts of logical reasoning Skills	PO5	6	Understand
CO3	Acquire satisfactory competency in use of reasoning and problem solving skills	PO5	6	Understand
CO4	Solve campus placements aptitude papers covering Quantitative Ability, Logical Reasoning Ability	PO5	6	Apply
CO5	Prepare themselves for various competitive exams and different placement aptitude test as well.	PO7	6	Apply

PEDAGOGY:

- Lecture, Group Discussion, LCD, Seminar & Case Study

UNIT: I **LECTURE HOURS: 6**

Arithmetical Ability: Numbers – Decimal fractions - HCF, LCM of numbers – Problems on numbers Simplifications, Average.

UNIT: II **LECTURE HOURS: 6**

Analytical Aptitude I: Percentage – Chain Rule - Ratio and Proportions – Profit and Loss.

UNIT: III **LECTURE HOURS: 6**

Analytical Aptitude II: Time and distance – Problems on trains – Boats and streams, Data Interpretation.

UNIT: IV **LECTURE HOURS: 6**

Analytical Aptitude III: Simple Interest – Compound Interest – Time and Work.

UNIT: V **LECTURE HOURS: 6**

Logical Reasoning: Seating Arrangement, Blood Relations, Number series, Picture Reasoning, Syllogism

ASSESSMENT

Objective type Question pattern – Paper pencil test.

REFERENCES

1. Agarwal.R.S– Quantitative Aptitude for Competitive Examinations, S.Chand and Limited, New Delhi, 2018.

2. AbhijitGuha, Quantitative Aptitude for Competitive Examinations, Tata McGraw Hill, 3 rd Edition, 2011.

3. Edgar Thrope, Test of Reasoning for Competitive Examinations, Tata McGraw Hill, 4 th Edition, 2012.

4. www.indiabix.com

SEMESTER –IV

Course Title **: Major Project and Viva Voce**	Course Code :**43R**
Semester: **IV**	Course Group : **PV**
Teaching Scheme in Hrs (L:T:P) : **0:0:30**	Credits : **15 Credits**
Map Code: NA	Total Contact Hours: **450**
CIA :**25 Marks**	SEE : **75Marks**
Programme:**MCA**	# - Semester End Exam

No.	Course Outcome	PSOs	Cl. Ses	CL
CO1	Demonstrate a sound technical knowledge of their selected project topic	PSO1, PSO4	90P	AP
CO2	Implement problem identification, formulation and solution	PSO1, PSO4	90P	AP
CO3	Solve industry solutions to complex problems utilising a systems approach	PSO1, PSO4	90P	AP
CO4	Practice system project using either web or mobile	PSO1, PSO4	90P	AP
CO5	Demonstrate the knowledge, skills and attitudes of a Full Stack Developer	PSO1, PSO4	90P	AP

PEDAGOGY:

- Lecture, Group Discussion, LCD, Seminar & Case Study

UNIT - I **90 Hours**

Understanding the Requirements: Applying Business Rules-(Requirements gathering)

UNIT - II **90 Hours**

Design: Page design –(User Interface)

UNIT - III **90 Hours**

SRS based Operations: Component and Content creation- Events, themes and styles, menus and navigation)

UNIT - IV **90 Hours**

Development: Configure, Connection and Creation-(Connector, API), Data Manipulation and Validation-(Test cases)

UNIT - V **90 Hours**

Implementation: Execution- (Hosting, Deployment)

REFERENCES:

1. https://courses.edx.org/courses/course-v1:HarvardX+CS50M+Mobile/course/
2. https://www.udemy.com/react-native-the-practical-guide/
3. https://www.udemy.com/react-native-advanced/
4. "Mastering React Native- A Beginners Guide" by Sufyan bin Uzayr CRC Press (Taylor & Francis Group) First Edition, 2023
5. Software Engineering by Ian Somerville publisher: Pearson Education Limited, 2016 Edition:10
6. BEGINNING Software Engineering by Rod Stephens Publisher: John Wiley & Sons, Inc, 2015
7. Project Management the Agile Way Making it Work in the Enterprise by John C. Good pasture, Publisher:J. Ross Publishing , 2nd Edition, 2016