



BMS INSTITUTE OF TECHNOLOGY AND MANAGEMENT
Yelahanka, Bengaluru – 560064.

Department of Computer Science and Engineering

COURSE FILE

WEB TECHNOLOGY

&

ITS APPLICATIONS

15CS71



BMS INSTITUTE OF TECHNOLOGY AND MANAGEMENT
Yelahanka, Bengaluru – 560064.

Department of Computer Science and Engineering

COURSE INFORMATION

Program	Computer Science & Engineering		
Course Code	15CS71	Course Title	Web technology & Its Applications
Course Coordinator Name	Shankar R	Academic Year	2018-19
Semester	VII	Section	A & B
Number of Lecture Hours/Week	04	Total Number of Lecture Hours	52
IA Marks	20	Exam Marks	80
Contact Number	9611616136	E-Mail	shankar@bmsit.in
Course Coordinator Signature			



BMS INSTITUTE OF TECHNOLOGY AND MANAGEMENT
Yelahanka, Bengaluru – 560064.

Department of Computer Science and Engineering

COURSE FILE CONTENTS

- 1. Calendar of Events**
- 2. Time Table**
- 3. About the Course**
- 4. Syllabus and Lesson Plan**
- 5. Course Outcomes**
- 6. CO – PO – PSO Mapping**
- 7. MOM with PAC (Gap Analysis, Students' Activity Plan)**
- 8. University Question Papers**
- 9. List of Students**
- 10. Internal Question Papers**
- 11. Scheme of Evaluation**
- 12. Result Analysis of Internal Tests**
- 13. OBE Activity Report with Feedback and Analysis**
- 14. Attainment Sheet with Articulation**
- 15. Sample Blue Books/Learning Materials/Assignments/PBL reports etc.**
- 16. Course End Report & Conclusion.**



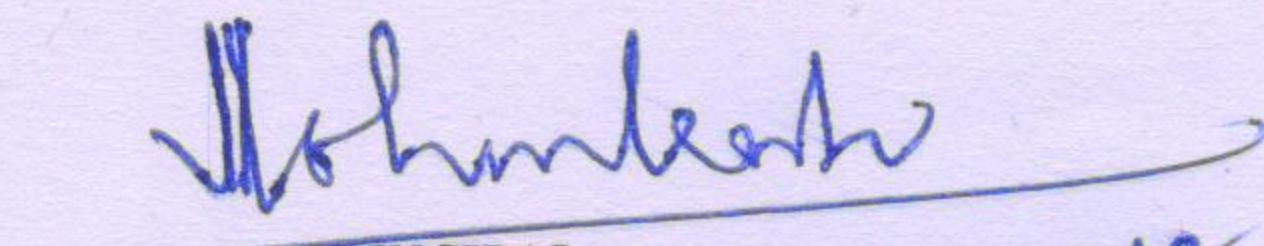
BMS Institute Of Technology and Management

Avalahalli, Yelahanka, Bengaluru – 560064

Calendar of Events 2018-19 (ODD Semester)

Month	Week	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Working Days	Events	Holidays
August	1st			1	2	3	4	5	4	1st Aug Commencement of III and V sem B.E/B.Tech 1st Aug Commencement of I, III sem MCA/M.Tech 2nd -7th Aug open courses for III & V sem	
	2nd	6	7	8	9	10	11	12	6	6th Aug Commencement of 7th sem B.E/B.Tech	
	3rd	13	14	15	16	17	18	19	5	13th Aug - 1st Sep Induction Programme for I Sem B.E/B.Tech/M.Tech/MCA	15th August Independence Day
	4th	20	21	22	23	24	25	26	5	20th Aug Project Orientation (only for final year), 25th Aug Academic Monitoring – 1 (Verification of academic documents)	22nd August Bakrid
	5th	27	28	29	30	31			6	29th Aug submissn of area of domain of final year project 1st sep Project Based Learning-1(Batch Creation) by Class Teacher.	
September	6th	3	4	5	6	7	8	9	6	4th sep Allocation of guides for Final year project, 4th Sep Project Based Learning-1(Guide Allocation)	
	7th	10	11	12	13	14	15	16	5	10th ,11th and 12th First Internal Assessment – Test1(III/V/VI), 8th Sept Academic Monitoring-2 (Verification of academic documents and counseling),	13th September Ganesh Chathurthi
	8th	17	18	19	20	21	22	23	5	17th Sep Project Based Learning - Synopsis Submission, 20th Sept test1 Marks entry to BIMS, 22nd Sept Dispatch of progress report (Sending SMS /Mail), 20th Sept Pre Assessment of final year project , 18th & 19th Tech Transform	21st September last day of Muharram
	9th	24	25	26	27	28	29	30	6	24th to 29th Sep Remedial classes for weak students 25th Sep Faculty appraisal by students – I, 24th Sept Academic Monitoring-3 (Verification of academic documents and counseling)	
	10th	1	2	3	4	5	6	7	5	3rd Oct Submission of Project Abstract 6th Oct Parent-Teacher Meeting	2nd October Gandhi Jayanthi,
October	11th	8	9	10	11	12	13	14	5	11th,12th, and 13th 2 nd Internal Test (III/V/VII), First Internal Test (I sem) 10th Oct Academic Monitoring-4 (Verification of academic documents and counseling)	8th October Mahalaya Amavasya
	12 th	15	16	17	18	19	20	21	4	17th Oct Review Meeting by SPARC 20th Sept test2 Marks entry to BIMS 20th Sept Dispatch of progress report (Sending SMS /Mail),	18th october Ayudha pooja 19th October Viyayadasami
	13th	22	23	24	25	26	27	28	5	22nd to 27th Oct Remedial classes for weak students, 25th Oct Academic Monitoring – 5 (Announcement of status of Attendance)	24th october valmiki Jayanthi
	14th	29	30	31		1	2	3	5	29th oct Open Day (College level Project/PBL Exhibition), 31st oct Final phase-1 Evaluation for final year	1st November Rajyotsava Day
	15th	5	6	7	8	9	10	11	4	10th Nov Academic Monitoring – 6 (Verification of academic documents and counseling),	6th November Naraka Chaturdashi 8th November Balipadyami Deepavalli
November	16th	12	13	14	15	16	17	18	6	12th, 13th & 14th Nov 3rd Internal Test (III/V/VII), 2nd Internal Test (I sem)	
	17th	19	20	21	22	23	24	25	5	19th Nov test3 Marks entry to BIMS, 22nd Nov Progress Report -Display of Final Attendance and IA Marks and Sending SMS / Mail),	21st November Id-e-Milad
	18th	26	27	28	29	30	1	2	5	30th Nov Last Working day for 3rd ,5th Sem B.E/B.Tech & 3rd sem M.Tech	26th November Kanakadasa jayanthi
	19th	3	4	5	6	7	8	9	6	4th Dec Last Working day for 7th Semester	
December	20th	10	11	12	13	14	15	16	6		
	21st	17	18	19	20	21	22	23	6	19th Dec Academic Monitoring – 7 (Verification of academic documents and counseling), 20th, 21st & 22nd Dec 3rd Internal Test (I sem)	
	22nd	24	25	26	27	28	29	30	5		25th December Christmas Eve
January		31	1	2	3	4	5	6	6		
		7	8	9	10	11	12	13	6	7th Jan Academic Monitoring – 8 (Verification of academic documents and counseling)	
		14	15	16	17				3	17th Jan Last Working day for 1st Semester	14th January Sankarantti
Total Number Of Working Days									91days for 3rd, 5th and 7th sem / 104 days for 1st sem		
VTU Examination									Practical Examinations: 03.12.2018 to 14.12.2018 (III & V Sem B.E/B.Tech). 03.12.2018 to 7.12.2018 (III & V Sem MCA) 06.12.2018 to 14.12.2018 (VII sem B.E/B.Tech) 21.01.2019 to 30.01.2019 (I Sem B.E/ B. Tech)		
									Theory Examinations: 17.12.2018 to 18.01.2019 (III,V & VI Sem B.E/B.Tech). 10.12.2018 to 28.12.2018 (III & V Sem MCA) 05.12.2018 to 22.12.2018 (III sem M.Tech) 04.02.2019 to 18.02.2019 (I Sem B.E/ B. Tech)		

*Student Centric Activity - Technical Talk/Seminar/Workshop/Quiz/HANDS-ON Session on Blended Learning (OBE Related activities) to be organized by the respective departments


PRINCIPAL
11.8.2018



BMS INSTITUTE OF TECHNOLOGY AND MANAGEMENT
YELAHANKA - BANGALORE - 64
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
INDIVIDUAL TIME TABLE FOR THE ACADEMIC YEAR 2018 – 19(ODD)

me: Mr. Shankar R	Subject: WEB	Semester: VII-A,B	Class room: BSN-CR-103, BSN-CR-104	W.E.F: 27-08-20
-------------------	--------------	-------------------	---------------------------------------	-----------------

	I 8.30 – 9.30	II 9.30 – 10.30	10.30 – 10.50	III 10.50 – 11.50	IV 11.50 – 12.50	12.50 – 1.45	V 1.45 – 2.40	VI 2.40 – 3.35	V 3.35 –	
MONDAY	WEB-A		TEA BREAK	WEB-B		LUNCH BREAK				
TUESDAY		WEB-B			WEB-A				WEB LAB-B1	
WEDNESDAY		WEB-A		WEB LAB-B2 (10.50AM-1.15PM)						
THURSDAY	WEB LAB-A2 (8.30AM-11.30AM)				WEB-A					
FRIDAY				WEB-B						
SATURDAY	WEB-B									

Total Workload(2T+3L)	17 Hours
-----------------------	----------

Time table officer

HoD

About the Course:

In order to make websites look and function a certain way, web developers utilize different languages. The three core languages that make up the World Wide Web are HTML5, CSS, and JavaScript.

In the IT world, the internet is an essential platform, whether its for developing or for consumer use. When developing a website, typically three main languages come into play. These languages are JavaScript, CSS, and HTML. HTML is the backbone of most webpages. Essentially, it is used to create the structure of how a specific website would look like, from the headings, to the paragraphs, the body, links, and even images.

Markup Languages

Markup languages are the languages in which the web is written. The most common markup language used is HTML, which uses tags to annotate text so that a computer can then manipulate the text. Most markup languages are human readable, and use annotations that are distinguishable from the annotated text. There are many different kinds of markups and languages, but all are consistent in the way in which they annotate documents.

Hypertext

Hypertext is defined as the arrangement of information inside a database that allows the user to receive information and to navigate from one document to another by clicking on highlighted words or pictures inside the primary document. Hypertext is the base of the World Wide Web, because it enables user to click on other links to get more information. Hypertext is a term used for all links, whether is appears as texts or other graphical part.

Hypertext Markup Language (HTML)

HTML is the conventional markup language used to create and edit web pages and web applications. HTML is used for creating the basic structure of a website. HTML consists of different elements preceded by an opening tag, <tag>, and a closing tag, </tag>. The content between the tags, <html> and </html>, is the content of the webpage. The content between the tags, <head> and </head>, is the title of the webpage. This text is displayed between the <title> and </title> tags. The content between the tags, <body> and </body>, is the main content of the webpage. The content can include links , paragraphs, headings, and various other elements.

Number of Lecture Hours/Week:03

Total Number of Lecture Hours:52

IA Marks: 20 **Exam Marks:**80

Delivery mode: Lecture, PPTs, Hands on.

Prerequisite for the program

Knowledge of computer components, basic programming skills.

Theory Syllabus

WEB TECHNOLOGY AND ITS APPLICATIONS [As per Choice Based Credit System (CBCS) scheme] (Effective from the academic year 2016 -2017) SEMESTER – VII

Subject Code	15CS71	IA Marks	20
Number of Lecture Hours/Week	4	Exam Marks	80
Total Number of Lecture Hours	50	Exam Hours	03

CREDITS – 04

Course objectives: This course will enable students to

- Illustrate the Semantic Structure of HTML and CSS
- Compose forms and tables using HTML and CSS
- Design Client-Side programs using JavaScript and Server-Side programs using PHP
- Infer Object Oriented Programming capabilities of PHP
- Examine JavaScript frameworks such as jQuery and Backbone

Module – 1	Teaching Hours
Introduction to HTML, What is HTML and Where did it come from?, HTML Syntax, Semantic Markup, Structure of HTML Documents, Quick Tour of HTML Elements, HTML5 Semantic Structure Elements, Introduction to CSS, What is CSS, CSS Syntax, Location of Styles, Selectors, The Cascade: How Styles Interact, The Box Model, CSS Text Styling	10 Hours
Module – 2	
HTML Tables and Forms, Introducing Tables, Styling Tables, Introducing Forms, Form Control Elements, Table and Form Accessibility, Microformats, Advanced CSS: Layout, Normal Flow, Positioning Elements, Floating Elements, Constructing Multicolumn Layouts, Approaches to CSS Layout, Responsive Design, CSS Frameworks.	10 Hours
Module – 3	
JavaScript: Client-Side Scripting, What is JavaScript and What can it do?, JavaScript Design Principles, Where does JavaScript Go?, Syntax, JavaScript Objects, The Document Object Model (DOM), JavaScript Events, Forms, Introduction to Server-Side Development with PHP, What is Server-Side Development, A Web Server's Responsibilities, Quick Tour of PHP, Program Control, Functions	10 Hours
Module – 4	
PHP Arrays and Superglobals, Arrays, \$_GET and \$_POST Superglobal Arrays, \$_SERVER Array, \$_Files Array, Reading/Writing Files, PHP Classes and Objects, Object-Oriented Overview, Classes and Objects in PHP, Object Oriented Design, Error Handling and Validation, What are Errors and Exceptions?, PHP Error Reporting, PHP Error and Exception Handling	10 Hours
Module – 5	
Managing State, The Problem of State in Web Applications, Passing Information via Query Strings, Passing Information via the URL Path, Cookies, Serialization, Session State, HTML5 Web Storage, Caching, Advanced JavaScript and jQuery, JavaScript Pseudo-Classes, jQuery Foundations, AJAX, Asynchronous File Transmission, Animation, Backbone MVC Frameworks, XML Processing and Web Services, XML Processing, JSON, Overview of Web Services.	10 Hours
Course outcomes: The students should be able to:	
<ul style="list-style-type: none"> <input type="checkbox"/> Adapt HTML and CSS syntax and semantics to build web pages. <input type="checkbox"/> Construct and visually format tables and forms using HTML and CSS <input type="checkbox"/> Develop Client-Side Scripts using JavaScript and Server-Side Scripts using PHP to generate and display the contents dynamically. <input type="checkbox"/> Appraise the principles of object oriented development using PHP <input type="checkbox"/> Inspect JavaScript frameworks like jQuery and Backbone which facilitates developerto focus on core features. 	

Question paper pattern:

The question paper will have TEN questions. There will be

TWO questions from each module.

Each question will have questions covering all the topics under a module.

The students will have to answer FIVE full questions, selecting ONE full question from each module.

TextBooks:

1. Randy Connolly, Ricardo Hoar, "Fundamentals of Web Development", 1 stEdition, Pearson Education India. (ISBN:978-9332575271)

Reference Books:

- 1) Robin Nixon, "Learning PHP, MySQL & JavaScript with jQuery, CSS and HTML5", 4 thEdition, O'Reilly Publications, 2015. (ISBN:978-9352130153)
- 2) Luke Welling, Laura Thomson, "PHP and MySQL Web Development", 5 th Edition, Pearson Education, 2016. (ISBN:978-9332582736)
- 3) Nicholas C Zakas, "Professional JavaScript for Web Developers", 3 rd Edition, Wrox/Wiley India, 2012. (ISBN:978-8126535088)
- 4) David Sawyer Mcfarland, "JavaScript & jQuery: The Missing Manual", 1 st Edition, O'Reilly/Shroff Publishers & Distributors Pvt Ltd, 2014 (ISBN:978- 9351108078)
- 5) Zak Ruvalcaba Anne Boehm, "Murach's HTML5 and CSS3" , 3rdEdition, Murachs/Shroff Publishers & Distributors Pvt Ltd, 2016. (ISBN:978-9352133246)

Lesson Plan

Lecture Hour	%	Module	Topics Covered	Text & Reference
1	20%	MODULE-I	Introduction to HTML, What is HTML and Where did it come from?	Text Book: 1.Randy Connolly, Ricardo Hoar, "Fundamentals of Web Development", 1st Edition, Pearson Education India.
2			HTML Syntax, Semantic Markup, Structure of HTML Documents	
3			Quick Tour of HTML Elements	
4			Quick Tour of HTML Elements	
5			HTML5 Semantic Structure Elements	
6			Introduction to CSS, What is CSS	
7			CSS Syntax, Location of Styles, Selectors	
8			The Cascade: How Styles Interact	
9			The Box Model	
10			CSS Text Styling	Reference Books: 1.Zak Ruvalcaba Anne Boehm, "Murach's HTML5 and CSS3", 3 rd Edition,Murachs/Shroff Publishers & Distributors Pvt Ltd, 2016. 2. Robin Nixon, "Learning PHP, MySQL &JavaScript with jQuery, CSS and HTML5", 4 th Edition, O'Reilly Publications, 2015. (ISBN:978-9352130153)
11	40%	MODULE-II	HTML Tables and Forms	Text Book: 1.Randy Connolly, Ricardo Hoar, "Fundamentals of Web Development", 1st Edition, Pearson Education India.
12			Introducing Tables, Styling Tables	
13			Introducing Forms, Form Control Elements	
14			Table and Form Accessibility	
15			Microformats	
16			Advanced CSS: Layout, Normal Flow	
17			Positioning Elements, Floating Elements	
18			Constructing Multicolumn Layouts	
19			Approaches to CSS Layout	
20			Responsive Design, CSS Frameworks	Reference Books: 1.Zak Ruvalcaba Anne Boehm, "Murach's HTML5 and CSS3", 3 rd Edition,Murachs/Shroff Publishers & Distributors Pvt Ltd,

				2016.
21			JavaScript: Client-Side Scripting	2. Robin Nixon, "Learning PHP, MySQL & JavaScript with jQuery, CSS and HTML5", 4 th Edition, O'Reilly Publications, 2015. (ISBN:978-9352130153)
22			What is JavaScript and What can it do?, JavaScript Design Principles	
23			Where does JavaScript Go?, Syntax	
24			JavaScript Objects	
25			The Document Object Model (DOM), JavaScript Events, Forms	
26			Introduction to Server-Side Development with PHP	
27			What is Server-Side Development, A Web Server's Responsibilities	
28			Quick Tour of PHP	
29			Program Control	
30			Functions	
31			PHP Arrays and Super globals	
32			Arrays, \$_GET and \$_POST Super global Arrays	
33			\$_SERVER Array, \$_Files Array, Reading/Writing Files	
34			PHP Classes and Objects, Object-Oriented Overview	
35			Classes and Objects in PHP	
36			Object Oriented Design	
37			Error Handling and Validation	
38			What are Errors and Exceptions?	
39			PHP Error Reporting	
40			PHP Error and Exception Handling	
41			Managing State, The Problem of State in Web Applications	
42			Passing Information via Query Strings	
43			Passing Information via the URL Path	
44			Cookies, Serialization, Session State	
45			HTML5 Web Storage, Caching, Advanced JavaScript and jQuery	
46			JavaScript Pseudo-Classes, jQuery Foundations	
47			AJAX, Asynchronous File Transmission, Animation	
48			Backbone MVC Frameworks,	
49			XML Processing and Web Services, XML Processing	
50			JSON, Overview of Web Services.	

Course Objectives

Illustrate the Semantic Structure of HTML and CSS
Compose forms and tables using HTML and CSS
Design Client-Side programs using JavaScript and Server-Side programs using PHP
Infer Object Oriented Programming capabilities of PHP
Examine JavaScript frameworks such as jQuery and Backbone

Course Outcomes (COs)

CO1	Discuss HTML, CSS, JS, PHP and their semantics to build web pages.
CO2	Apply the underlying web technologies to meet the end user's demands.
CO3	Analyze the static web contents and dynamic web contents of world wide web.
CO4	Design Client-Side Scripts and Server-Side Scripts to generate and display the contents dynamically.
CO5	Investigate a case study which will be given and relate to the underlying core concepts.
CO6	Develop useful applications illustrating the concepts learnt.

Programme outcomes (POs):

1.	Engineering knowledge: Apply the knowledge of Mathematics, Science, Engineering fundamentals and an engineering specialization to the solution of complex engineering problems
2.	Problem analysis: Identify, formulate, review research literature, and analyse complex Engineering problems reaching substantiated conclusions using first principles of mathematics, Natural sciences and engineering sciences
3.	Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4.	Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the Information to provide valid conclusions
5.	Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern Engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
6.	The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7.	Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for Sustainable development
8.	Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9.	Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings
10.	Communication: Communicate effectively on complex engineering activities with the engineering Community and with society at large, such as, being able to comprehend and write effective reports And design documentation, make effective presentations, and give and receive clear instructions.
11.	Project management and finance: Demonstrate knowledge and understanding of the Engineering and management principles and apply these to one's own work, as a member and Leader in a team, to manage projects and in multidisciplinary environments.
12	Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes (PSOs):

PSO1	Apply design and development principles in the construction of software systems of varying complexity.
PSO2	Analyze the problem and identify computing requirements appropriate to its solution.

CO - PO - PSO Mapping																
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	Delivery Mechanism	
CO1: Discuss HTML, CSS, JS, PHP and their semantics to build web pages.															Lecture, PPT	
CO2: Apply the underlying web technologies to meet the end user's demands.		3													Lecture, PPT	
CO3: Analyze the static web contents and dynamic web contents of world wide web.			3												Lecture, Hands on, PPT	
CO4: Design Client-Side Scripts and Server-Side Scripts to generate and display the contents dynamically.				3										2	1	Lecture, Hands on, PPT
CO5: Investigate a case study which will be given and relate to the underlying core concepts.					3											Lecture, Hands on, PPT

CO6: Develop useful applications illustrating the concepts learnt.					2	3					3	3	2	2	3	2	HANDS ON
---	--	--	--	--	---	---	--	--	--	--	---	---	---	---	---	---	----------

Articulation - Mapping of COs with POs & PSOs

CO1 is not mapped to any PO since CO1 discusses about the understanding levels.

CO2 is mapped to PO1 - Apply level - K3.

CO3 is mapped to PO2 - Analyze level - K4.

CO4 is mapped to PO3 - Design level - K5.

CO5 is mapped to PO4 - Case Study - K4.

CO6 is mapped accordingly to PBL - Develop level - K5.

CO4 is mapped to PSO1 and PSO2 - Analyze level - K4 & Develop level - K5.

Gap identified based on CO-PO mapping

- Gaps are identified from PO5 to PO12
- PBL projects to be executed by students in teams by using modern tools and a report has to be presented on the same.

Course End Report

TARGET				LEVEL
60%	STUDENTS MUST SCORE	60%	& ABOVE	3-High
55%	STUDENTS MUST SCORE	60%	& ABOVE	2-Moderate
50%	STUDENTS MUST SCORE	60%	& ABOVE	1-Low

COURSE OUTCOMES	ATTAINMENT LVL-I A	ATTAINMENT LVL-UNIVERSITY	OVERALL ATTAINMENT LVL
CO1	3.00	1.00	1.80
CO2	3.00	1.00	1.80
CO3	3.00	1.00	1.80
CO4	3.00	1.00	1.80
CO5: CASE STUDY	3.00		3.00
CO6-OBE	3.00		3.00

CO4: Design Client-Side Scripts and Server-Side Scripts to generate and display the contents dynamically.	1.80	0	0	1.8	0	0	0	0	0	0	0	0	0	0	1.2	0.6
CO5: Investigate a case study which will be given and relate to the underlying core concepts.	3.00	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0
CO6: Develop useful applications illustrating the concepts learnt.	3.00	0	0	0	2	3	0	0	0	3	3	2	2	3	2	2
SUM	1.8	1.8	1.8	5	3	0	0	0	3	3	2	2	4.2	2.6		
ATTAINMENT	1.8	1.8	1.8	2.5	3	0	0	0	3	3	2	2	2.1	1.3		

UNIVERSITY EXAM CONTRIBUTION

Attainment of all COs is 1. (meaning, more than 50% of the students were able to score more than 60% of the marks in university examination)

There was a gap for PO5 to PO12. So, an activity of mini project (PBL) was conducted. The outcome is: Students got more exposure towards the web application development via modern tool usage, team involvement, project maintenance & lifelong learning, also all the mini projects were evaluated for 5 marks - contributing to direct attainment.

CONCLUSION

High Attainment: PO4, PO5, PO9, PO10 (≈ 3)

Moderate Attainment: PO1, PO2, PO3, PO11, PO12, PSO1 (≈ 2)

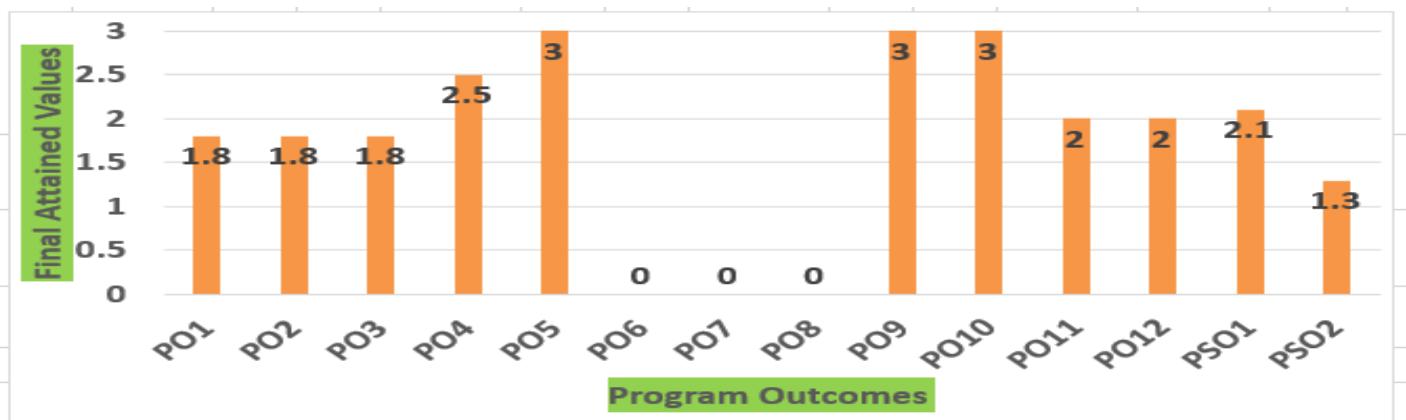
Low Attainment: PSO2 (≈ 1)

Not Attained: PO6, PO7, PO8. (to try to attain in the next academic year)

University Exam Marks need to be improved. All the Moderate and Low attained POs to be improved.

FINAL ATTAINMENT ANALYSIS - WEB - 15CS71 - 2018-19 - 2015 Batch

PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
1.8	1.8	1.8	2.5	3	0	0	0	3	3	2	2	2.1	1.3





Minutes of Meeting with PAC

Date	16.08.2018	Location	CSE DEPT
Time	10:40 AM	Module Coordinator	Anand R
Courses	Current Course Coordinator	Previous course coordinator	
Web Technology and its Applications - 15CS71	Shankar R	--	
Sl.No	Discussion	Action By/ Responsible	Remarks
1	Agenda: Course Outcomes, CO – PO – PSO Mapping, Gap Identification for Web Technology and its Applications – 15CS71.		
2	<p>The Course Outcomes (COs) Web Technology and its Applications – 15CS71 given in university curriculum are as follows:</p> <p>CO1: Adapt HTML and CSS syntax and semantics to build web pages.</p> <p>CO2: Construct and visually format tables and forms using HTML and CSS</p> <p>CO3: Develop Client-Side Scripts using JavaScript and Server-Side Scripts using PHP to generate and display the contents dynamically. CO4: Appraise the principles of object oriented development using PHP</p> <p>CO5: Inspect JavaScript frameworks like jQuery and Backbone which facilitates developer to focus on core features.</p> <p>The Observations of the committee are as follows:</p> <ol style="list-style-type: none">1. The skill words are missing in all CO, so skill words should be included in CO for better mapping with PO.2. CO1, CO2 and CO3 include the basic fundamental outcomes.3. End of the course, what are the skills students will be able to acquire pertaining to the POs mapped? The skill acquired should be to develop Web applications for the given requirements.4. Therefore, it was decided to change the COs and include advanced topics and skill words like apply, analyse design and investigate in CO2, CO3 and CO4, to give equal weightage to all the topics in the syllabus. <p>CO - PO - PSO Mapping</p> <p>CO1 is not mapped to any PO since CO1 discusses about the understanding levels.</p> <p>CO2 is mapped to PO1 – Apply level – K3.</p> <p>CO3 is mapped to PO2 – Analyze level – K4.</p> <p>CO4 is mapped to PO3 – Design level – K5.</p> <p>CO5 is mapped to PO4 - Case Study – K4.</p> <p>CO6 is mapped accordingly to PBL – Develop level – K5.</p> <p>CO4 is mapped to PSO1 and PSO2 - Analyze level – K4 & Develop level – K5.</p> <p>New redefined COs</p> <p>CO1: Discuss HTML, CSS, JS, PHP and their semantics to build web pages.</p> <p>CO2: Apply the underlying web technologies to meet the end user's demands.</p> <p>CO3: Analyze the static web contents and dynamic web contents of world</p>	Course Coordinator	

	<p>wide web.</p> <p>CO4: Design Client-Side Scripts and Server-Side Scripts to generate and display the contents dynamically.</p> <p>CO5: Investigate a case study which will be given and relate to the underlying core concepts.</p> <p>CO6: Develop useful applications illustrating the concepts learnt.</p> <p>Gap identified based on CO-PO mapping</p> <ul style="list-style-type: none"> • Gaps are identified from PO5 to PO12 • PBL projects to be executed by students in teams by using modern tools and a report has to be presented on the same. 		
3	<p>Student centric activity planned to bridge the gap</p> <p>It was decided to conduct the PBL for the entire class with 2 students in a team and CO6 is mapped to PO4, PO5, PO9, PO10, PO11, PO12 and PSO1, PSO2.</p>		

CO-PO mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1: Discuss HTML, CSS, JS, PHP and their semantics to build web pages.														
CO2: Apply the underlying web technologies to meet the end user's demands.	3													
CO3: Analyze the static web contents and dynamic web contents of world wide web.		3												
CO4: Design Client-Side Scripts and Server-Side Scripts to generate and display the contents dynamically.			3										2	1
CO5: Investigate a case study which will be given and relate to the underlying core concepts.				3										
CO6: Develop useful applications illustrating the concepts learnt.				2	3				3	3	2	2	3	2

Course Coordinator	Module Coordinator	Program Coordinator	HOD

#

SL. NO.	USN	NAME OF THE CANDIDATE
1.	1BY15CS002	AAYUSH NANDKEOLYAR
2.	1BY15CS003	AAYUSH SAPKOTA
3.	1BY15CS004	ABHISHEK
4.	1BY15CS005	ABHISHEK B
5.	1BY15CS006	ADARSH KUMAR SAH
6.	1BY15CS007	ADITYA SUBRAYA HEGDE
7.	1BY15CS008	AMOGH G
8.	1BY15CS009	ANANYA
9.	1BY15CS010	ANKIT AGRAWAL
10.	1BY15CS011	ANUSHREE GUDOOR
11.	1BY15CS013	ARUNDHATI S SHANBHAG
12.	1BY15CS015	BHANU PRATAP SINGH
13.	1BY15CS016	BHARGAV SAGIRAJU
14.	1BY15CS017	BHAVESH M
15.	1BY15CS018	BHAVYA SAH
16.	1BY15CS020	CHANDRA SHEKAR
17.	1BY15CS021	CHINMAYI RAJASHEKAR HEGDE
18.	1BY15CS023	DEEPTHA S
19.	1BY15CS024	DISHI TIWARI
20.	1BY15CS025	DIVYASHREE G
21.	1BY15CS027	GODI LEHYA REDDY
22.	1BY15CS028	HARSHA B S
23.	1BY15CS029	HARSHITH CHAUDHARY
24.	1BY15CS030	HARSHITH T R
25.	1BY15CS032	JIVESH BODH
26.	1BY15CS033	JYOTHI S
27.	1BY15CS034	K VARUN KARANTH
28.	1BY15CS035	KANYA KRISHI
29.	1BY15CS037	KARAN V
30.	1BY15CS038	KARTHIK L N
31.	1BY15CS039	KAVYA M
32.	1BY15CS040	KETAN YADAV
33.	1BY15CS041	KUMAR GAURAV RAJ
34.	1BY15CS042	KUMAR ROHIT
35.	1BY15CS043	KUNAL KUMAR
36.	1BY15CS044	LAKSHMI HARSHITHA M
37.	1BY15CS045	MADHUKAR BALAKRISHNA SHETTY
38.	1BY15CS046	MANISH K
39.	1BY15CS047	MANOJ RAJARAM HEGDE
40.	1BY15CS048	MAYANK KUSHAL
41.	1BY15CS049	MEGHANA M
42.	1BY15CS050	MEGHNA SHARMA
43.	1BY15CS051	MOHAMMED DAANIYAAL
44.	1BY15CS052	MOHAMMED ZAMAN
45.	1BY14CS019	YAMINI B. T
46.	1BY16CS401	AMAR. R
47.	1BY16CS403	CHETHANA D.N
48.	1BY16CS404	DIVYA SHREE G
49.	1BY16CS406	MAMATHA K
50.	1BY16CS408	PRIYANKA G
51.	1BY16CS410	RAVISHA S.V

HOD
Dept. of CSE

BMS INSTITUTE OF TECHNOLOG & MANAGEMENT, YELAHANKA, BANGALORE-64

Batch: 2015 - 16

Branch: CSE

Sem: VII

' B ' Section

AY: 2018-19

Dt: 08-08-2018

SL. NO.	USN	NAME OF THE CANDIDATE
1.	1BY15CS053	MOHEIN RANJITKAR
2.	1BY15CS055	NANDINI V
3.	1BY15CS056	NIKHIL A R
4.	1BY15CS057	NIKIL N ACHAR
5.	1BY15CS058	NITIN MISHRA
6.	1BY15CS059	NITTURI POORNIMA NICHALA
7.	1BY15CS060	P VIGNESH
8.	1BY15CS061	PAVAN HOYSAL L S
9.	1BY15CS062	POOJITHA K C
10.	1BY15CS063	PRAFULLA SREEVALSAN
11.	1BY15CS065	PREETHAM T
12.	1BY15CS066	PRERAN DEVAIAH
13.	1BY15CS067	R S VARUN
14.	1BY15CS068	RADHIKA B RAMAN
15.	1BY15CS069	RAKESH SAH
16.	1BY15CS070	RISHABH VERMA
17.	1BY15CS072	RUCHITA KRISHNAN IYER
18.	1BY15CS074	SABJAN P
19.	1BY15CS075	SACHIN KUMAR S
20.	1BY15CS076	SAHIL VERMA
21.	1BY15CS077	SAMAR KHAN
22.	1BY15CS078	SANDHYA M
23.	1BY15CS080	SHALINI B N
24.	1BY15CS082	SHREYA S
25.	1BY15CS083	SHUBHI SHARMA
26.	1BY15CS085	SUHIEL BASTOLA
27.	1BY15CS086	SUNANDA BANIK
28.	1BY15CS088	SUPRIYA S WARAD
29.	1BY15CS089	SUSHIMTHA N
30.	1BY15CS090	T SRIPAD
31.	1BY15CS092	TANVI BHAKTA
32.	1BY15CS093	TEJAS VAIDYA
33.	1BY15CS095	TUSHAR SAHAY
34.	1BY15CS096	ULLAS V B
35.	1BY15CS097	UMESH DAFTARI
36.	1BY15CS098	V ARCHITH
37.	1BY15CS100	VIGHNESHA RAUT
38.	1BY15CS102	VISHNU MENON
39.	1BY15CS103	VISHWANANTH BHATH S
40.	1BY15CS106	SAURABH BANGAR
41.	1BY15CS108	THATODE SAI SURYA ABHIS
42.	1BY15CS109	RAHUL RAJ K R
43.	1BY15CS110	BHAVYA K SHETH
44.	1BY15CS111	ANAM KHAN
45.	1BY15CS112	AKSHAY A BHANDARKAR
46.	1BY15CS113	ADITYA S
47.	1BY15CS114	VINITH V B
48.	1BY14CS113	RITWIK MOHANTY
49.	1BY16CS409	RANJITHA R
50.	1BY16CS411	SACHIN T
51.	1BY16CS412	SANDEEP M. N
52.	1BY16CS415	SINCHANA M K
53.	1BY16CS417	VINOD G
54.	1BY15CS116	VINAY H R

HOD
Dept of CSE

Total =105



BMS INSTITUTE OF TECHNOLOGY AND MANAGEMENT
YELAHANKA - BANGALORE - 64

Department of Computer Science & Engineering

FIRST INTERNAL EXAMINATION, SEPTEMBER 2018 (CBCS)

Subject: Web Technology and its Applications	Subject Code: 15CS71	Semester : VII CSE 'A'&'B'
Max. Marks : 30	Date: 10-09-2018	Time: 09:30 AM to 11:00 AM

Answer FIVE full questions choosing 3 questions from Part A. Part B is compulsory.

Part A

1	Illustrate the following with example. i) span ii) div iii) aside iv) Pseudo-classes in CSS	6 Marks	(CO2) (PO1,K2)
---	--	------------	-------------------

OR

2	Illustrate the following with example a)float property in CSS b)form c)input d) a	6 Marks	(CO2) (PO1,K2)
---	--	------------	-------------------

3	Explain Responsive Design and setting viewports with examples.	6 Marks	(CO1) (K2)
---	--	------------	---------------

OR

4	Discuss Box model and Illustrate how to use external style sheet having an unordered list, table, borders, padding, color, and .	6 Marks	(CO2) (PO1,K2)
---	--	------------	-------------------

5	Discuss various CSS Selectors with examples.	6 Marks	(CO1) (K2)
---	--	------------	---------------

OR

6	Demonstrate Relative, Absolute, Z-index positioning with example in CSS.	6 Marks	(CO2) (PO1,K2)
---	--	------------	-------------------

Part B

7	Develop a HTML5 document to describe ordered list of your five favorite movies. You should surround some text about that movie's image. Each element of the list must have a nested list of at least two actors in your favorite movies.	6 Marks	(CO4) (PO3,K5)
8	We discussed BMSIT&M's timetable structure in class which includes open-course, proctor meet, co-curricular activities, tutorial class information etc. Based on this case study, Design a web page to create this time table for your section of 7 th CSE.	6 Marks	(CO5) (PO4,K5)

Course Outcomes (COs)

Students will be able to

CO1	Discuss HTML, CSS, JS, PHP and their semantics to build web pages.
CO2	Apply the underlying web technologies to meet the end user's demands.
CO3	Analyze the static web contents and dynamic web contents of world wide web.
CO4	Design Client-Side Scripts and Server-Side Scripts to generate and display the contents dynamically.
CO5	Investigate a case study which will be given and relate to the underlying core concepts.
CO6	Develop useful applications illustrating the concepts learnt.

Bloom's Taxonomy

K1- Remembering, K2 - Understanding, K3 - Applying, K4 - Analyzing, K5 - Evaluating, K6 - Creating

Faculty Signature

Module Coordinator

PAC

HoD, CSE



BMS INSTITUTE OF TECHNOLOGY AND MANAGEMENT
YELAHANKA - BANGALORE - 64

Department of Computer Science & Engineering

FIRST INTERNAL EXAMINATION SCHEME, SEPTEMBER 2018 (CBCS)

Subject: Web Technology and its Applications	Subject Code: 15CS71	Semester : VIIICSE 'A'&'B'
Max. Marks : 30	Date: 10- 09 - 2018	Time: 09:30 AM to 11:00 AM

Answer FIVE full questions choosing 3 questions from Part A. Part B is compulsory.

Part A

1	<p>Illustrate the following with example.</p> <p>i) span- The <code></code> tag is used to group inline-elements in a document. The <code></code> tag provides no visual change by itself. The <code></code> tag provides a way to add a hook to a part of a text or a part of a document. <code><p>My mother has blue eyes.</p></code></p> <p>ii) div- The <code><div></code> tag defines a division or a section in an HTML document. The <code><div></code> element is often used as a container for other HTML elements to style them with CSS or to perform certain tasks with JavaScript. <code><div style="background-color:lightblue"><h3>This is a heading</h3><p>This is a paragraph.</p></div></code></p> <p>iii) aside- The <code><aside></code> tag defines some content aside from the content it is placed in. The aside content should be related to the surrounding content. <code><p>My family and I visited The Epcot center this summer.</p></code></p> <p><code><aside><h4>Epcot Center</h4><p>The Epcot Center is a theme park in Disney World, Florida.</p></aside></code></p> <p>iv) Pseudo-classes in CSS- A pseudo-class is used to define a special state of an element.</p> <p>For example, it can be used to:</p> <p>Style an element when a user mouses over it Style visited and unvisited links differently Style an element when it gets focus</p> <pre>/* unvisited link */ a:link { color: #FF0000; } /* visited link */ a:visited { color: #00FF00; }</pre>	1.5 * 4 =6 Marks	(CO2) (PO1,K2)

	<pre> /* mouse over link */ a:hover { color: #FF00FF; } /* selected link */ a:active { color: #0000FF; } </pre>		
OR			
	<p>Illustrate the following with example</p> <p>a) float property in CSS- The float property specifies how an element should float.</p> <p>Note: Absolutely positioned elements ignores the float property!</p> <p>Note: Elements after a floating element will flow around it.</p> <pre> img { float: right; } </pre> <p>b) form - The HTML <code><form></code> element defines a form that is used to collect user input:</p> <pre> <form> First name:
 <input type="text" name="firstname">
 Last name:
 <input type="text" name="lastname"> </form> </pre>		
2	<p>c) input - <code><input type="text"></code> defines a one-line input field for text input:</p> <pre> <form> First name:
 <input type="text" name="firstname">
 Last name:
 <input type="text" name="lastname"> </form> </pre> <p><code><input type="radio"></code> defines a radio button. Radio buttons let a user select ONE of a limited number of choices:</p> <pre> <form> <input type="radio" name="gender" value="male" checked> Male
 <input type="radio" name="gender" value="female"> Female
 <input type="radio" name="gender" value="other"> Other </form> </pre>	$1.5 * 4 = 6$ Marks	(CO1) (K2)

	<p>d) a - The <a> tag defines a hyperlink, which is used to link from one page to another.</p> <p>The most important attribute of the <a> element is the href attribute, which indicates the link's destination.</p> <p>By default, links will appear as follows in all browsers:</p> <ul style="list-style-type: none"> • An unvisited link is underlined and blue • A visited link is underlined and purple • An active link is underlined and red <p>Visit W3Schools.com!</p>	
--	---	--

3	<p>Explain Responsive Design and setting viewports with examples.</p> <p>In a responsive design, the page "responds" to changes in the browser size that go beyond the width scaling of a liquid layout. One of the problems of a liquid layout is that images and horizontal navigation elements tend to take up a fixed size, and when the browser window shrinks to the size of a mobile browser, liquid layouts can become unusable. In a responsive layout, images will be scaled down and navigation elements will be replaced as the browser shrinks</p> <p>There are four key components that make responsive design work. They are:</p> <ol style="list-style-type: none"> 1. Liquid layouts 2. Scaling images to the viewport size 3. Setting viewports via the <meta>tag 4. Customizing the CSS for different viewports using media queries <p>Responsive designs begin with a liquid layout, that is, one in which most elements have their widths specified as percentages. Making images scale in size is actually quite straightforward, in that you simply need to specify the following rule:</p> <pre>img { max-width: 100%; }</pre> <p>Of course this does not change the downloaded size of the image; it only shrinks or expands its visual display to fit the size of the browser window, never expanding beyond its actual dimensions. More sophisticated responsive designs will serve different sized images based on the viewport size.</p> <p>Setting Viewports</p> <p>A key technique in creating responsive layouts makes use of the ability of current mobile browsers to shrink or grow the web page to fit the width of the screen. If you have ever used a modern mobile browser, you may have been surprised to see how the web page was scaled to fit into the small screen of the browser. The way this works is the mobile browser renders the page on a canvas called the viewport. On iPhones, for instance, the viewport width is 980 px, and then that viewport is scaled to fit the current width of the device (which can change with orientation and with newer versions that have more physical pixels in the screen). The mobile Safari browser introduced the viewport <meta>tag as a way for developers to control the size of that initial viewport.</p> <pre><html> <head> <meta name="viewport" content="width=device-width" /></pre>	<p>3*2= 6 Marks</p> <p>(CO2) (PO1,K2)</p>
---	---	---

	By setting the viewport as in this listing, the page is telling the browser that no scaling is needed, and to make the viewport as many pixels wide as the device screen width. This means that if the device has a screen that is 320 px wide, the viewport width will be 320 px; if the screen is 480 px (for instance, in landscape mode), then the viewport width will be 480 px.		
--	---	--	--

OR

	Discuss Box model and Illustrate how to use external style sheet having an unordered list, table, borders, padding, color, and .		
4	<pre><!DOCTYPE html> <html> <head> <link rel='stylesheet' type='text/css' href='mystyle.css'> </head> <body> <h2>Demonstrating the Box Model</h2> <p>The CSS box model is essentially a box that wraps around every HTML element. It consists of: borders, padding, margins, and the actual content.</p> <div>This text is the actual content of the box. We have added a 25px padding, 25px margin and a 25px green border. Ut enim ad minim veniam, quis nostrud exercitation ullamcolaboris nisi ut aliquip ex ea commodoconsequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteursint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</div> </body> </html></pre> <p>mystyle.css</p> <pre>div { background-color: lightgrey; width: 300px; border: 25px solid green; padding: 25px; margin: 25px; } span { color: red; }</pre>	6 Marks	(CO2) (PO1,K2)

	Discuss various CSS Selectors with examples.		
5	<p>Element Selectors</p> <p>Element selectors select all instances of a given HTML element. You can select all elements by using the universal element selector, which is the * (asterisk) character. You can select a group of elements by separating the different element names with commas. This is a sensible way to reduce the size and complexity of your CSS files, by combining multiple identical rules into a single rule.</p>	1.5 * 4 = 6 Marks	(CO1) (K2)

```

/* commas allow you to group selectors */
p, div, aside {
margin: 0;
padding: 0;
}
/* the above single grouped selector is equivalent to the
following: */
p {
margin: 0;
padding: 0;
}
div {
margin: 0;
padding: 0;
}
aside {
margin: 0;
padding: 0;
}

```

Class Selectors

A **class selector** allows you to simultaneously target different HTML elements regardless of their position in the document tree. If a series of HTML elements have been labeled with the same class attribute value, then you can target them for styling by using a class selector, which takes the form: period (.) followed by the class name.

```

<head>
<title>Share Your Travels </title>
<style>
.first {
font-style: italic;
color: red;
}
</style>
</head>
<body>
<h1 class="first">Reviews</h1>
<div>
<p class="first">By Ricardo on <time>September 15, 2015</time></p>
<p>Easy on the HDR buddy.</p>
</div>
<hr/>
<div>
<p class="first">By Susan on <time>October 1, 2015</time></p>
<p>I love Central Park.</p>
</div>
<hr/>

```

Id Selectors

An **id selector** allows you to target a specific element by its id attribute regardless of its type or position. If an HTML element has been labeled with an id attribute, then you can target it for styling by using an id selector, which takes the form: pound/hash (#) followed by the id name.

```

head lang="en">
<meta charset="utf-8">
<title>Share Your Travels -- New York - Central Park</title>

```

```

<style>
#latestComment {
font-style: italic;
color: red;
}
</style>
</head>
<body>
<h1>Reviews</h1>
<div id="latestComment">
<p>By Ricardo on <time>September 15, 2015</time></p>
<p>Easy on the HDR buddy.</p>
</div>
<hr/>
<div>
<p>By Susan on <time>October 1, 2015</time></p>
<p>I love Central Park.</p>
</div>
<hr/>
</body>

```

Attribute Selectors

An **attribute selector** provides a way to select HTML elements either by the presence of an element attribute or by the value of an attribute. This can be a very powerful technique, but because of uneven support by some of the browsers, not all web authors have used them. Attribute selectors can be a very helpful technique in the styling of hyperlinks and images. For instance, perhaps we want to make it more obvious to the user when a pop-up tooltip is available for a link or image. We can do this by using the following attribute selector:

```

[title] { ... }

<head lang="en">
<meta charset="utf-8">
<title>Share Your Travels</title>
<style>
[title] {
cursor: help;
padding-bottom: 3px;
border-bottom: 2px dotted blue;
text-decoration: none;
}
</style>
</head>
<body>
<div>
<imgsrc="images/flags/CA.png" title="Canada Flag" />
<h2><a href="countries.php?id=CA" title="see posts from Canada">
Canada</a>
</h2>
<p>Canada is a North American country consisting of . . . </p>
<div>
<imgsrc="images/square/6114907897.jpg"
title="At top of Sulphur Mountain" />
<imgsrc="images/square/6592317633.jpg"
title="Grace Presbyterian Church" />
<imgsrc="images/square/6592914823.jpg"

```

```

title="Calgary Downtown" />
</div>
</div>
</body>

```

This will match any element in the document that has a title attribute.

OR

Demonstrate Relative, Absolute, Z-index positioning with example in CSS.

Relative Positioning

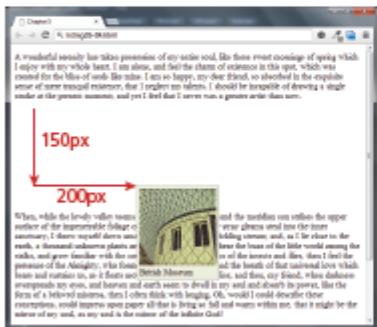
In **relative positioning** an element is displaced out of its normal flow position and moved relative to where it would have been placed. When an element is positioned relatively, it is displaced out of its normal flow position and moved relative to where it would have been placed. The other content around the relatively positioned element “remembers” the element’s old position in the flow; thus the space the element would have occupied is preserved.



```

<p>A wonderful serenity has taken possession of my ...</p>
<figure>
  
  <figcaption>British Museum</figcaption>
</figure>
<p>When, while the lovely valley ...</p>

```



```

figure {
  border: 1px solid #A8A8A8;
  background-color: #EDEDED;
  padding: 5px;
  width: 150px;
  position: relative;
  top: 150px;
  left: 200px;
}

```

6

2*3=6
Marks (CO2)
(PO1,K2)

FIGURE 5.4 Relative positioning

As you can see in Figure 5.4, the original space for the positioned `<figure>` element is preserved, as is the rest of the document’s flow. As a consequence, the repositioned element now overlaps other content: that is, the `<p>` element following the `<figure>` element does not change to accommodate the moved `<figure>`.

Absolute Positioning

When an element is positioned absolutely, it is removed completely from normal flow. Thus, unlike with relative positioning, space is not left for the moved element, as it is no longer in the normal flow. Its position is moved in relation to its container block. In the example shown in Figure, the container block is the `<body>` element. Like with the relative positioning example, the moved block can now overlap content in the underlying normal flow

 <p>A wonderful serenity has taken possession of my entire soul. like these sweet mornings of spring, which I enjoy with my whole heart. I am alone, and feel the charm of existence in this spot, which was allotted for the� of souls like mine. I am so happy, my dearest friend, as I should be in the exquisite sense of mere temporal existence, that I ought not to single out the present moment, and yet I find that I never write a greater sentence than this.</p> <p>When, while the lovely valley seems with vapour wrapped around me, and the meridian sun softens the upper surface of the implemental foliage of my trees, and but a few steps distance over, into the same tranquillity, I have myself drawn across the tall grass by the winding stream, and, as I sit alone by the water, a few simple substances plants are scattered by me; when I have due leave of the time, I sit here, for solitude, and for contemplation, and for the contemplation of the works of the seasons and of God, then I feel the presence of the Almighty, who formed me in his own image, and the breath of that universal love which bears and sustains us, as it flows around us in all the works of creation.</p> <p>British Museum</p> <p>When, while the lovely valley seems with vapour wrapped around me, and the meridian sun softens the upper surface of the implemental foliage of my trees, and but a few steps distance over, into the same tranquillity, I have myself drawn across the tall grass by the winding stream, and, as I sit alone by the water, a few simple substances plants are scattered by me; when I have due leave of the time, I sit here, for solitude, and for contemplation, and for the contemplation of the works of the seasons and of God, then I feel the presence of the Almighty, who formed me in his own image, and the breath of that universal love which bears and sustains us, as it flows around us in all the works of creation.</p> <p>British Museum</p> <pre> <p>A wonderful serenity has taken possession of my ..</p> <figure> <figcaption>British Museum</figcaption> </figure> <p>When, while the lovely valley ..</p> </pre> <p>Figure {</p> <pre> margin: 0; border: 1px solid #A8A8A8; background-color: #EDEDED; padding: 5px; width: 150px; position: absolute; top: 150px; left: 200px; } </pre>			
Part B	<p>Develop a HTML5 document to describe ordered list of your five favorite movies. You should surround some text about that movie's image. Each element of the list must have a nested list of at least two actors in your favorite movies.</p>	7	6 Marks (CO4) (PO3,K5)

	 Actor 1 Actor 2 </body> </html>		
8	<p>We discussed BMSIT&M's timetable structure in class which includes open-course, proctor meet, co-curricular activities, tutorial class information etc. Based on this case study, Design a web page to create this time table for your section of 7th CSE.</p> <pre> <html> <head> <title>time table</title> </head> <bodybgcolor="skyblue"> <H1><FONTCOLOR="DARKCYAN"><CENTER>COLLEGE TIME TABLE</H1> <tableborder="2"cellspacing="3"align="center"> <tr> <tdalign="center"> <td>8:30-9:30 <td>9:30-10:30 <td>10:3-11:30 <td>11:30-12:30 <td>12:30-2:00 <td>2:00-3:00 <td>3:00-4:00 <td>4:00-5:00 </td> </tr> <tr> <tdalign="center">MONDAY <tdalign="center">---<tdalign="center"><fontcolor="blue">SUB1
 <tdalign="center"><fontcolor="pink">SUB2
 <tdalign="center"><fontcolor="red">SUB3
 <trowspan="6"align="center">L
U
N
C
H <tdalign="center"><fontcolor="maroon">SUB4
 <tdalign="center"><fontcolor="brown">SUB5
 <tdalign="center">counselling class </td> </tr> <tr> <tdalign="center">TUESDAY <tdalign="center"><fontcolor="blue">SUB1
 <tdalign="center"><fontcolor="red">SUB2
 <tdalign="center"><fontcolor="pink">SUB3
 <tdalign="center">--- <tdalign="center"><fontcolor="orange">SUB4
 <tdalign="center"><fontcolor="maroon">SUB5
 <tdalign="center">library </td> </tr> <tr> <tdalign="center">WEDNESDAY <tdalign="center"><fontcolor="pink">SUB1
 <tdalign="center"><fontcolor="orange">SUB2
 <tdalign="center"><fontcolor="brown">SWA
 <tdalign="center">--- <tdcolspan="3"align="center"><fontcolor="green"> lab </td> </tr> </pre>	6	Marks (CO5) (PO4,K5)

```

</tr>
<tr>
<tdalign="center">THURSDAY
<tdalign="center">SUB1<br>
<tdalign="center"><fontcolor="brown">SUB2<br>
<tdalign="center"><fontcolor="orange">SUB3<br>
<tdalign="center">---
<tdalign="center"><fontcolor="blue">SUB4<br>
<tdalign="center"><fontcolor="red">SUB5<br>
<tdalign="center">library
</tr>
<tr>
<tdalign="center">FRIDAY
<tdalign="center"><fontcolor="orange">SUB1<br>
<tdalign="center"><fontcolor="maroon">SUB2<br>
<tdalign="center"><fontcolor="blue">SUB3<br>
<tdalign="center">---
<tdalign="center"><fontcolor="pink">SUB4<br>
<tdalign="center"><fontcolor="brown">SUB5<br>
<tdalign="center">library
</tr>
<tr>
<tdalign="center">SATURDAY
<tdalign="center"><fontcolor="red">SUB1<br>
<tdcolspan="3" align="center">seminar
<tdalign="center"><fontcolor="pink">SUB4<br>
<tdalign="center"><fontcolor="brown">SUB5<br>
<tdalign="center">library
</tr>
</body>
</html>

```

Course Outcomes (COs)

Students will be able to

CO1	<i>Discuss HTML, CSS, JS, PHP and their semantics to build web pages.</i>
CO2	<i>Apply the underlying web technologies to meet the end user's demands.</i>
CO3	<i>Analyze the static web contents and dynamic web contents of world wide web.</i>
CO4	<i>Design Client-Side Scripts and Server-Side Scripts to generate and display the contents dynamically.</i>
CO5	<i>Investigate a case study which will be given and relate to the underlying core concepts.</i>
CO6	<i>Develop useful applications illustrating the concepts learnt.</i>

Bloom's Taxonomy

K1- Remembering, K2 - Understanding, K3 - Applying, K4 - Analyzing, K5 - Evaluating, K6 - Creating

Faculty Signature

Module Coordinator

PAC

HoD, CSE



BMS INSTITUTE OF TECHNOLOGY AND MANAGEMENT
YELAHANKA - BANGALORE - 64

Department of Computer Science & Engineering

SECOND INTERNAL EXAMINATION, OCTOBER 2018 (CBCS)

Subject: Web Technology and its Applications	Subject Code: 15CS71	Semester : VIICSE 'A'&'B'
Max. Marks : 30	Date: 13- 10 - 2018	Time: 09:30 AM to 11:00 AM

Answer FIVE full questions choosing 3 questions from Part A. Part B is compulsory.

Part A

1	Write a JavaScript to design a simple calculator to perform the following operations:sum, product, difference and quotient.	6 Marks	(CO2) (PO1,K3)
---	---	------------	-------------------

OR

2	Illustrate the various Array methods in JavaScript with examples.	6 Marks	(CO2) (PO1,K3)
---	---	------------	-------------------

3	Explain JavaScript Design Principles.	6 Marks	(CO1) (K2)
---	---------------------------------------	------------	---------------

OR

4	Breakdown DOM with its methods and illustrate modifying its Elements?	6 Marks	(CO3) (PO2,K4)
---	---	------------	-------------------

5	Discuss \$_GET and \$_POST Super global Arrays.	6 Marks	(CO1) (K2)
---	---	------------	---------------

OR

6	Demonstrate Parameter passing techniques in PHP.	6 Marks	(CO2) (PO1,K3)
---	--	------------	-------------------

Part B

7	Imagine you have a blog. Develop a PHP program to keep track of the number of visitors visiting your blog and display the count of visitors, with proper headings.	6 Marks	(CO4) (PO3,K5)
---	--	------------	-------------------

8	We discussed BMSIT&M's students' video resume uploading to LinkedIn website for better job perspectives. Based on this case study, Develop a webpage for this and discuss how to handle the file upload in PHP and check for errors for better validations w.r.t File size restrictions.	6 Marks	(CO5) (PO4,K5)
---	--	------------	-------------------

Course Outcomes (COs)

Students will be able to

CO1	Discuss HTML, CSS, JS, PHP and their semantics to build web pages.
CO2	Apply the underlying web technologies to meet the end user's demands.
CO3	Analyze the static web contents and dynamic web contents of world wide web.
CO4	Design Client-Side Scripts and Server-Side Scripts to generate and display the contents dynamically.
CO5	Investigate a case study which will be given and relate to the underlying core concepts.
CO6	Develop useful applications illustrating the concepts learnt.

Bloom's Taxonomy

K1- Remembering, K2 - Understanding, K3 - Applying, K4 - Analyzing, K5 - Evaluating, K6 - Creating

Faculty Signature

Module Coordinator

PAC

HoD, CSE



BMS INSTITUTE OF TECHNOLOGY AND MANAGEMENT
YELAHANKA - BANGALORE - 64

Department of Computer Science & Engineering

SECOND INTERNAL EXAMINATION SCHEME, OCTOBER 2018 (CBCS)

Subject: Web Technology and its Applications	Subject Code: 15CS71	Semester : VII CSE 'A'&'B'
Max. Marks : 30	Date: 13- 10 - 2018	Time: 09:30 AM to 11:00 AM

Answer FIVE full questions choosing 3 questions from Part A. Part B is compulsory.

Part A

1	<p>Write a JavaScript to design a simple calculator to perform the following operations: sum, product, difference and quotient.</p> <pre><!DOCTYPE HTML> <html> <head> <style> table, td, th { border: 1px solid black; width: 33%; text-align: center; background-color: DarkGray; border-collapse: collapse; } table { margin: auto; } input { text-align: right; } </style> <script type="text/javascript"> function calc(clicked_id) { var val1 = parseFloat(document.getElementById("value1").value); var val2 = parseFloat(document.getElementById("value2").value); if(isNaN(val1) isNaN(val2)) alert("ENTER VALID NUMBER"); else if(clicked_id=="add") document.getElementById("answer").value=val1+val2; else if(clicked_id=="sub") document.getElementById("answer").value=val1-val2; else if(clicked_id=="mul") document.getElementById("answer").value=val1*val2; else if(clicked_id=="div") document.getElementById("answer").value=val1/val2; } function cls() { value1.value="0"; value2.value="0"; }</pre>	6 Marks	(CO2) (PO1,K3)
---	--	------------	---------------------------

```

        answer.value="";
    }

</script>
</head>
<body>
<table>
<tr><th colspan="4"> SIMPLE CALCULATOR </th></tr>
<tr><td>value1</td><td><input type="text" id="value1" value="0"/></td>
<td>value2</td><td><input type="text" id="value2" value="0"/></td></tr>
<tr><td><input type="button" value="Addition" id = "add" onclick="calc(this.id)"/></td>
<td><input type="button" value="Subtraction" id = "sub" onclick="calc(this.id)"/></td>
<td><input type="button" value="Multiplication" id = "mul" onclick="calc(this.id)"/></td>
<td><input type="button" value="Division" id = "div" onclick="calc(this.id)"/></td></tr>
<tr><td>Answer:</td><td><input type="text" id="answer" value="" disabled/></td>
<td colspan="2"><input type="button" value="CLEAR ALL" onclick="cls()"/></td>

</tr>
</table>
</body>
</html>

```

OR

Illustrate the various Array methods in JavaScript with examples.

Arrays are one of the most used data structures, and they have been included in JavaScript as well. In practice, this class is defined to behave more like a linked list than it can be resized dynamically, but the implementation is browser specific, meaning the efficiency of insert and delete operations is unknown.

Arrays will be the first objects we will examine. Objects can be created using the new syntax and calling the object constructor. The following code creates a new, empty array named greetings:

```
var greetings = new Array();
```

To initialize the array with values, the variable declaration would look like the following:

```
var greetings = new Array("Good Morning", "Good Afternoon");
```

or, using the square bracket notation:

```
var greetings = ["Good Morning", "Good Afternoon"];
```

2

1.5*4
= 6
Marks
(CO2)
(PO1,K3)

Accessing and Traversing an Array

To access an element in the array you use the familiar square bracket notation from Java and C-style languages, with the index you wish to access inside the brackets.

```
alert ( greetings[0] );
```

Modifying an Array

To add an item to an existing array, you can use the push method.

```
greetings.push("Good Evening");
```

Math

The **Math class** allows one to access common mathematical functions and common values quickly in one place. This static class contains methods such as max(), min(), pow(), sqrt(), and exp(), and trigonometric functions such as sin(), cos(), and arctan().

	<pre>Math.PI// 3.141592657 Math.sqrt(4); //square root of 4 is 2. Math.random(); // random number between 0 and 1</pre>		
--	---	--	--

String

The **String class** has already been used without us even knowing it. That is because it is core to communicating with the user. Since it is so common, shortcuts have been defined for creating and concatenating strings. While one can use the new syntax to create a String object, it can also be defined using quotes as follows:

```
var greet = new String("Good");// long form constructor
var greet = "Good";// shortcut constructor
```

A common need is to get the length of a string. This is achieved through the `length` property (just as in arrays).

```
alert(greet.length);// will display "4"
```

Another common way to use strings is to concatenate them together. Since this is so common, the `+` operator has been overridden to allow for concatenation in place.

```
varstr = greet.concat("Morning");// Long form concatenation
varstr = greet + "Morning";// + operator concatenation
```

Date

While not critical to JavaScript, the Date class is yet another helpful included object you should be aware of. It allows you to quickly calculate the current date or create date objects for particular dates. To display today's date as a string, we would simply create a new object and use the `toString()` method.

```
var d = new Date();
// This outputs Today is Mon Nov 12 2012 15:40:19 GMT-0700
alert("Today is "+ d.toString());
```

Explain JavaScript Design Principles.

These principles increase the quality and reusability of the code while making it easier to understand, and hence more maintainable.

Layers

When designing software to solve a problem, it is often helpful to abstract the solution a little bit to help build a cognitive model in your mind that you can then implement. Perhaps the most common way of articulating such a cognitive model is via the term **layer**. In object-oriented programming, a software **layer** is a way of conceptually grouping programming classes that have similar functionality and dependencies. Common software design layer names include:

- **Presentation layer.** Classes focused on the user interface.
- **Business layer.** Classes that model real-world entities, such as customers, products, and sales.
- **Data layer.** Classes that handle the interaction with the data sources.

3

2*3=6
Marks (CO1)
(K2)

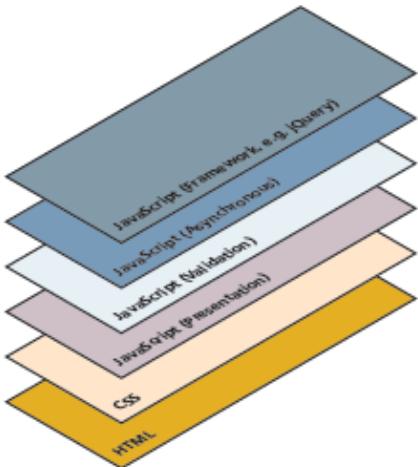


FIGURE 6.8 JavaScript layers

Presentation Layer

This type of programming focuses on the display of information. JavaScript can alter the HTML of a page, which results in a change, visible to the user. These presentation layer applications include common things like creating, hiding, and showing divs, using tabs to show multiple views, or having arrows to page through resultsets. This layer is most closely related to the user experience and the most visible to the end user.

Validation Layer

JavaScript can also be used to validate logical aspects of the user's experience. This could include, for example, validating a form to make sure the email entered is valid before sending it along. It is often used in conjunction with the presentation layer to create a coherent user experience, where a message to the presentation layer highlights bad fields. Both layers exist on the client machine, although the intention is to prevalidate forms before making transmissions back to the server.

Asynchronous Layers

Normally, JavaScript operates in a synchronous manner where a request sent to the server requires a response before the next lines of code can be executed. During the wait between request and response the browser sits in a loading state and only updates upon receiving the response. In contrast, an asynchronous layer can route requests to the server in the background. In this model, as certain events are triggered, the JavaScript sends the HTTP requests to the server, but while waiting for the response, the rest of the application functions normally, and the browser isn't in

a loading state. When the response arrives JavaScript will (perhaps) update a portion of the page. Asynchronous layers are considered advanced versions of the presentation and validation layers above

Users without JavaScript

Too often website designers believe (erroneously) that users without JavaScript are somehow relics of a forgotten age, using decades-old computers in a bomb shelter somewhere philosophically opposed to updating their OS and browsers and therefore not worth worrying about. Nothing could be more of a straw man argument. Users have a myriad of reasons for not using JavaScript, and that includes some of the most important clients, like search engines. A client may not have JavaScript

because they are a web crawler, have a browser plug-in, are using a text browser, or are visually impaired.

- **Web crawler.** A web crawler is a client running on behalf of a search engine to download your site, so that it can eventually be featured in their search results. These automated software agents do not interpret JavaScript, since it is costly, and the crawler cannot see the enhanced look anyway.

■ **Browser plug-in.** A browser plug-in is a piece of software that works within the browser, that might interfere with JavaScript. There are many uses of JavaScript that are not desirable to the end user. Many malicious sites use JavaScript to compromise a user's computer, and many ad networks deploy advertisements using JavaScript. This motivates some users to install plug-ins that stop JavaScript execution. An ad-blocking plug-in, for example, may filter JavaScript scripts that include the word *ad*, so a script named **advanced.js** would be blocked inadvertently.

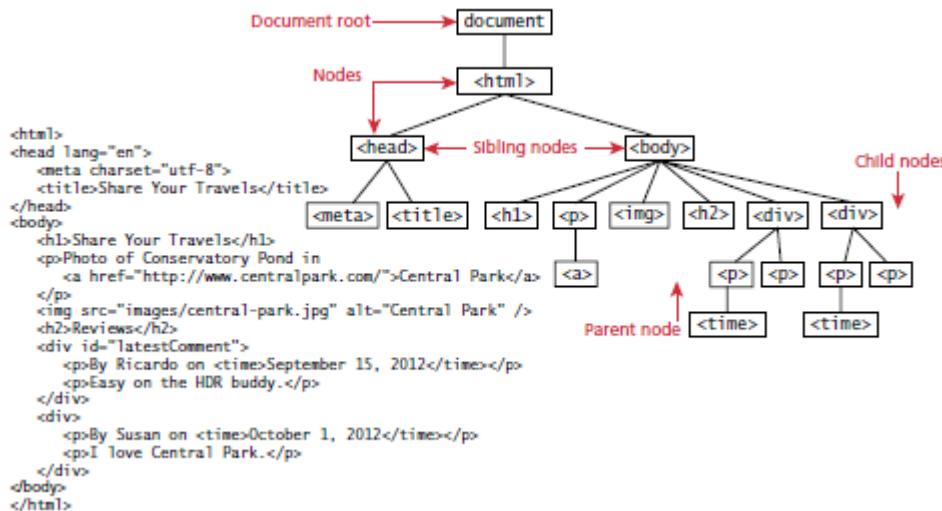
■ **Text-based client.** Some clients are using a text-based browser. Text-based browsers are widely deployed on web servers, which are often accessed using a command-line interface. A website administrator might want to see what an HTTP GET request to another server is returning for testing or support purposes.

■ **Visually disabled client.** A visually disabled client will use special web browsing software to read the contents of a web page out loud to them. These specialized browsers do not interpret JavaScript, and some JavaScript on sites is not accessible to these users. Designing for these users requires some extra considerations, with lack of JavaScript being only one of them.

OR

Breakdown DOM with its methods and how do you modify its Elements?

JavaScript is almost always used to interact with the HTML document in which it is contained. As such, there needs to be some way of programmatically accessing the elements and attributes within the HTML. This is accomplished through a programming interface (API) called the **Document Object Model (DOM)**.



4

2*3=6
Marks
**(CO3)
(PO2,K4)**

Nodes

In the DOM, each element within the HTML document is called a **node**. If the DOM is a tree, then each node is an individual branch. There are element nodes, text nodes, and attribute nodes, as shown in Figure

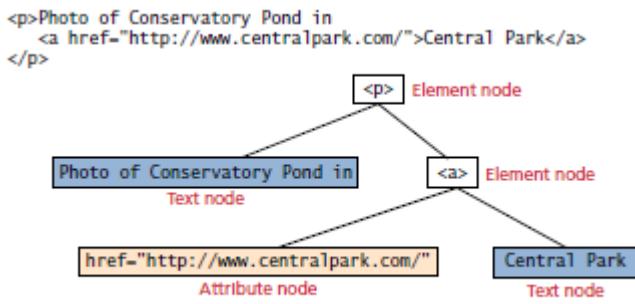


FIGURE 6.18 DOM nodes

Document Object

The **DOM document object** is the root JavaScript object representing the entireHTML document. It contains some properties and methods that we will use extensivelyin our development and is globally accessible as document.

Method	Description
<code>createAttribute()</code>	Creates an attribute node
<code>createElement()</code>	Creates an element node
<code>createTextNode()</code>	Creates a text node
<code>getElementById(id)</code>	Returns the element node whose id attribute matches the passed id parameter
<code>getElementsByName(name)</code>	Returns a NodeList of elements whose tag name matches the passed name parameter

TABLE 6.4 Some Essential Document Object Methods

Modifying a DOM Element

Using the DOM document and HTML DOM element objects, we can do exactly that using the `innerHTML`property

```
var latest = document.getElementById("latestComment");
var oldMessage = latest.innerHTML;
latest.innerHTML = oldMessage + "<p>Updated this div with JS</p>";
```

Now the HTML of our document has been modified to reflect that change.

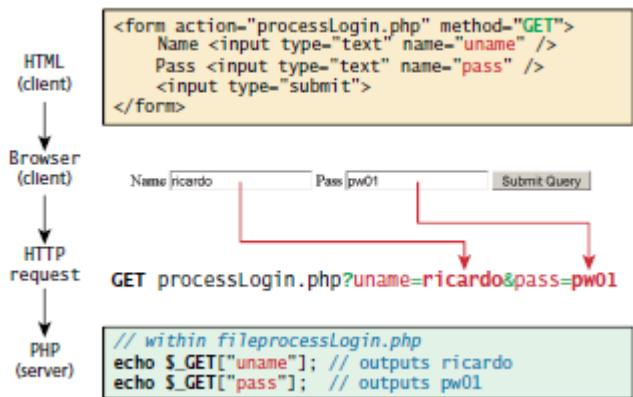
```
<div id="latestComment">
<p>By Ricardo on <time>September 15, 2012</time></p>
<p>Easy on the HDR buddy.</p>
<p>Updated this div with JS</p>
</div>
```

Discuss \$_GET and \$_POST Superglobal Arrays.

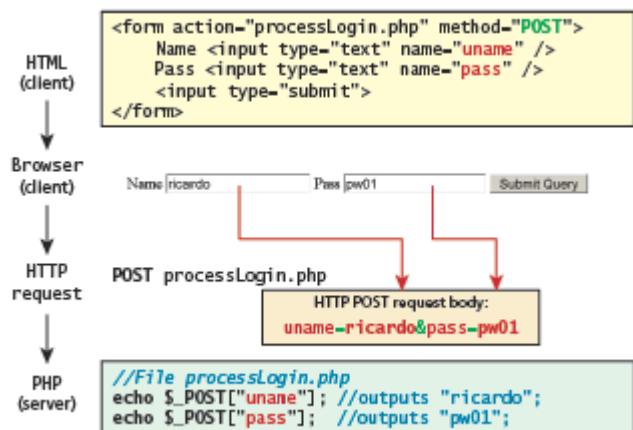
The `$_GET` and `$_POST` arrays are the most important superglobal variables in PHP since they allow the programmer to access data sent by the client in a query string. As you will recall, an HTML form (or an HTML link) allows a client to send data to the server. That data is formatted such that each value is associated with a name defined in the form. If the form was submitted using an HTTP GET request, then the resulting URL will contain the data in the query string. PHP will populate the superglobal `$_GET` array using the contents of this query string in the URL.

5

2*3=6
Marks
(CO1)
(K2)



If the form was sent using HTTP POST, then the values would not be visible in the URL, but will be sent through HTTP POST request body. From the PHP programmer's perspective, almost nothing changes from a GET data post except that those values and keys are now stored in the `$_POST` array. This mechanism greatly simplifies accessing the data posted by the user, since you need not parse the querystring or the POST request headers.



OR

Demonstrate Parameter passing techniques in PHP.

Parameters are the mechanism by which values are passed into functions, and there are some complexities that allow us to have multiple parameters, default values, and to pass objects by reference instead of value.

```

function getNiceTime($showSeconds)
{
if ($showSeconds==true)
return date("H:i:s");
else
return date("H:i");
}
    
```

Thus to call our function, you can now do it in two ways:

```

echo getNiceTime(1); // this will print seconds
echo getNiceTime(0); // will not print seconds
    
```

Parameter Default Values

In PHP you can set **parameter default values** for any parameter in a function. However, once

6

2*3=6
(CO2)
(PO1,K3)

you start having default values, all subsequent parameters must also have defaults.

```
function getNiceTime($showSeconds=1){  
if ($showSeconds==true)  
return date("H:i:s");  
else  
return date("H:i");  
}
```

Passing Parameters by Reference

By default, arguments passed to functions are **passed by value** in PHP. This means that PHP passes a copy of the variable so if the parameter is modified within the function, it does not change the original.

```
function changeParameter($arg) {  
    $arg += 300;  
    echo "<br/>arg=". $arg;  
}  
  
$initial = 15;  
echo "<br/>initial=". $initial; // output: initial-15  
changeParameter($initial); // output: arg-315  
echo "<br/>initial=". $initial; // output: initial-15
```

LISTING 8.17 Passing a parameter by value

Like many other programming languages, PHP also allows arguments to functions to be **passed by reference**, which will allow a function to change the contents of a passed variable. A parameter passed by reference points the local variable to the same place as the original, so if the function changes it, the original variable is changed as well. The mechanism in PHP to specify that a parameter is passed by reference is to add an ampersand (&) symbol next to the parameter name in the function declaration.

```
function changeParameter(&$arg) {  
    $arg += 300;  
    echo "<br/>arg=". $arg;  
}  
  
$initial = 15;  
echo "<br/>initial=". $initial; // output: initial-15  
changeParameter($initial); // output: arg-315  
echo "<br/>initial=". $initial; // output: initial-315
```

LISTING 8.18 Passing a parameter by reference

Part B

Imagine you have a blog. Develop a PHP program to keep track of the number of visitors visiting your blog and display the count of visitors, with proper headings.

7

```
<?php  
  
print "<h3><center> Welcome to my Blog!</center></h3>";
```

6
Marks

(CO4)
(PO3,K5)

	<pre> \$fname="counter.txt"; \$fp = fopen(\$fname,"r"); \$hits= fscanf(\$fp,"%d"); fclose(\$fp); \$hits[0]++; \$fp = fopen(\$fname,"w"); fprintf(\$fp,"%d",\$hits[0]); fclose(\$fp); print "Total number of views: ".\$hits[0]; ?> </pre>		
8	<p>We discussed BMSIT&M's students' video resume uploading to LinkedIn website for better job perspectives. Based on this case study, Develop a webpage for this and discuss how to handle the file upload in PHP and check for errors for better validations w.r.t File size restrictions.</p> <pre> HTML (client) ↓ Browser (client) C:\Users\ricardo\Pictures\Sample1.png Browse... Submit Query ↓ HTTP request ↓ PHP (server) </pre> <pre> <form enctype='multipart/form-data' method='post' action='upFile.php'> <input type='file' name='file1' /> <input type='submit' /> </form> POST upFile.php HTTP POST multipart/form-data file1=@C:\Users\ricardo\Pictures\Sample1.png;type=image/png echo \$_FILES["file1"]["name"] // "Sample1.png" echo \$_FILES["file1"]["type"] // "image/png" echo \$_FILES["file1"]["tmp_name"] // "/tmp/phpJ08pVh" echo \$_FILES["file1"]["error"] // 0 echo \$_FILES["file1"]["size"] // 1219038 </pre>	2*3=6 (CO5) (PO4,K5) Marks	
	<h3>Checking for Errors</h3> <p>For every uploaded file, there is an error value associated with it in the <code>\$_FILE\$</code> array. The error values are specified using constant values, which resolve to integers. The value for a successful upload is <code>UPLOAD_ERR_OK</code>, and should be looked for before proceeding any further.</p> <h3>File Size Restrictions</h3> <p>Some scripts limit the file size of each upload. There are many reasons to do so, and ideally you would prevent the file from even being transmitted in the first place if it is too large. There are three main mechanisms for maintaining uploaded file size restrictions: via HTML in the input form, via JavaScript in the input form, and via PHP coding.</p> <pre> <form enctype='multipart/form-data' method='post'> <input type="hidden" name="MAX_FILE_SIZE" value="1000000" /> <input type='file' name='file1' /> <input type='submit' /> </form> </pre>		

Course Outcomes (COs)

Students will be able to

CO1 Discuss HTML, CSS, JS, PHP and their semantics to build web pages.

CO2	<i>Apply the underlying web technologies to meet the end user's demands.</i>
CO3	<i>Analyze the static web contents and dynamic web contents of world wide web.</i>
CO4	<i>Design Client-Side Scripts and Server-Side Scripts to generate and display the contents dynamically.</i>
CO5	<i>Investigate a case study which will be given and relate to the underlying core concepts.</i>
CO6	<i>Develop useful applications illustrating the concepts learnt.</i>
Bloom's Taxonomy	
K1- Remembering, K2 - Understanding, K3 - Applying, K4 - Analyzing, K5 - Evaluating, K6 - Creating	

Faculty Signature

Module Coordinator

PAC

HoD, CSE



BMS INSTITUTE OF TECHNOLOGY AND MANAGEMENT
YELAHANKA - BANGALORE - 64

Department of Computer Science & Engineering

THIRD INTERNAL EXAMINATION, NOVEMBER 2018 (CBCS)

Subject: Web Technology and its Applications	Subject Code: 15CS71	Semester : VII CSE 'A'&'B'
Max. Marks : 30	Date: 19- 11 - 2018	Time: 09:30 AM to 11:00 AM

Answer FIVE full questions choosing 3 questions from Part A. Part B is compulsory.

Part A

1	Compare Serialization and Deserialization with example snippet.	6 Marks	(CO3) (PO2,K4)
---	---	------------	-------------------

OR

2	Analyze HTML5 Web Storage with code to read and write web storage.	6 Marks	(CO3) (PO2,K4)
---	--	------------	-------------------

3	Illustrate various jQuery Selectors with examples.	6 Marks	(CO2) (PO1,K3)
---	--	------------	-------------------

OR

4	Analyze AJAX requests via jqXHR object.	6 Marks	(CO3) (PO2,K4)
---	---	------------	-------------------

5	Analyze SOAP services and REST services.	6 Marks	(CO3) (PO2,K4)
---	--	------------	-------------------

OR

6	Demonstrate XML processing in PHP through XMLReader and SimpleXML techniques.	6 Marks	(CO2) (PO1,K3)
---	---	------------	-------------------

Part B

7	Imagine you have to animate an element using jQuery. Develop a simple contact form/script that builds and shows a clickable email link when you click the email icon. Also, discuss about the usage of animate() with a step function to do CSS3 rotation.	6 Marks	(CO4) (PO3,K5)
---	--	------------	-------------------

8	Analyze the document that was shared (http://backbonejs.org). Based on this case study, Design a Backbone.js Collection defined to hold Photo Albums and build a custom view object for the same.	6 Marks	(CO5) (PO4,K5)
---	---	------------	-------------------

Course Outcomes (COs)

Students will be able to

CO1	Discuss HTML, CSS, JS, PHP and their semantics to build web pages.
CO2	Apply the underlying web technologies to meet the end user's demands.
CO3	Analyze the static web contents and dynamic web contents of world wide web.
CO4	Design Client-Side Scripts and Server-Side Scripts to generate and display the contents dynamically.
CO5	Investigate a case study which will be given and relate to the underlying core concepts.
CO6	Develop useful applications illustrating the concepts learnt.

Bloom's Taxonomy

K1- Remembering, K2 - Understanding, K3 - Applying, K4 - Analyzing, K5 - Evaluating, K6 - Creating

Course Coordinator	Module Coordinator	Programme Coordinator	HoD
--------------------	--------------------	-----------------------	-----



BMS INSTITUTE OF TECHNOLOGY AND MANAGEMENT
YELAHANKA - BANGALORE - 64

Department of Computer Science & Engineering

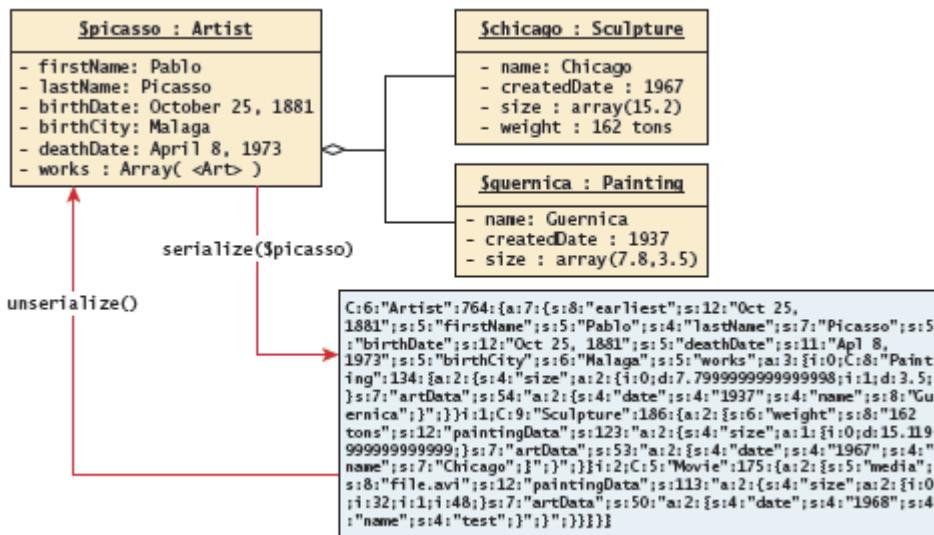
THIRD INTERNAL EXAMINATION SOLUTIONS, NOVEMBER 2018 (CBCS)

Subject: Web Technology and its Applications	Subject Code: 15CS71	Semester : VIICSE 'A'&'B'
Max. Marks : 30	Date: 19- 11 - 2018	Time: 09:30 AM to 11:00 AM
<i>Answer FIVE full questions choosing 3 questions from Part A. Part B is compulsory.</i>		

Part A

Compare Serialization and Deserialization with example snippet.

Serialization is the process of taking a complicated object and reducing it down to zeros and ones for either storage or transmission. Later that sequence of zeros and ones can be reconstituted into the original object as illustrated in Figure.



1

6
Marks
(CO3)
(PO2,K4)

In PHP objects can easily be reduced down to a binary string using the `serialize()` function. The resulting string is a binary representation of the object and therefore may contain unprintable characters. The string can be reconstituted back into an object using the `unserialize()` method.

```

interface Serializable {
    /* Methods */
    public function serialize();
    public function unserialize($serialized);
}

```

While arrays, strings, and other primitive types will be serializable by default, classes of our own creation must implement the `Serializable` interface shown in program, which requires adding implementations for `serialize()` and `unserialize()` to any class that implements this interface.

The following code shows how the `Artist` class must be modified to implement the `Serializable` interface by adding the `implements` keyword to the class definition and adding implementations for the two methods.

```

class Artist implements Serializable {
    /**
     * Implement the Serializable interface methods
     */
    public function serialize() {
        // use the built-in PHP serialize function
        return serialize(
            array("earliest" -> self::$earliestDate,
                  "first" -> $this->firstName,
                  "last" -> $this->lastName,
                  "bdate" -> $this->birthDate,
                  "ddate" -> $this->deathDate,
                  "bcity" -> $this->birthCity,
                  "works" -> $this->artworks
            );
    }
}

public function unserialize($data) {
    // use the built-in PHP unserialize function
    $data = unserialize($data);
    self::$earliestDate = $data['earliest'];
    $this->firstName = $data['first'];
    $this->lastName = $data['last'];
    $this->birthDate = $data['bdate'];
    $this->deathDate = $data['ddate'];
    $this->birthCity = $data['bcity'];
    $this->artworks = $data['works'];
}

//...
}

```

OR

Analyze HTML5 Web Storage with code to read and write web storage.

Web storage is a new JavaScript-only API introduced in HTML5. It is meant to be a replacement (or perhaps supplement) to cookies, in that web storage is managed by the browser; unlike cookies, web storage data is not transported to and from the server with every request and response. In addition, web storage is not limited to the 4K size barrier of cookies; the W3C recommends a limit of 5MB but browsers are allowed to store more per domain.

Writing web storage

The following code illustrates the JavaScript code for writing information to web storage. Do note that it is *not* PHP code that interacts with the web storage mechanism but JavaScript. As demonstrated in the listing, there are two ways to store values in web storage: using the `setItem()` function, or using the property shortcut (e.g., `sessionStorage.FavoriteArtist`).

```

<form ... >
    <h1>Web Storage Writer</h1>
    <script language="javascript" type="text/javascript">

        if (typeof (localStorage) === "undefined" ||
            typeof (sessionStorage) === "undefined") {
            alert("Web Storage is not supported on this browser...");
        }
        else {
            sessionStorage.setItem("TodaysDate", new Date());
            sessionStorage.FavoriteArtist = "Matisse";

            localStorage.UserName = "Ricardo";
            document.write("web storage modified");
        }
    </script>
    <p><a href="WebStorageReader.php">Go to web storage reader</a></p>
</form>

```

Reading web storage

The following code illustrates that the process of reading from web storage is equally straightforward.

2

6 Marks (CO3) (PO2,K4)

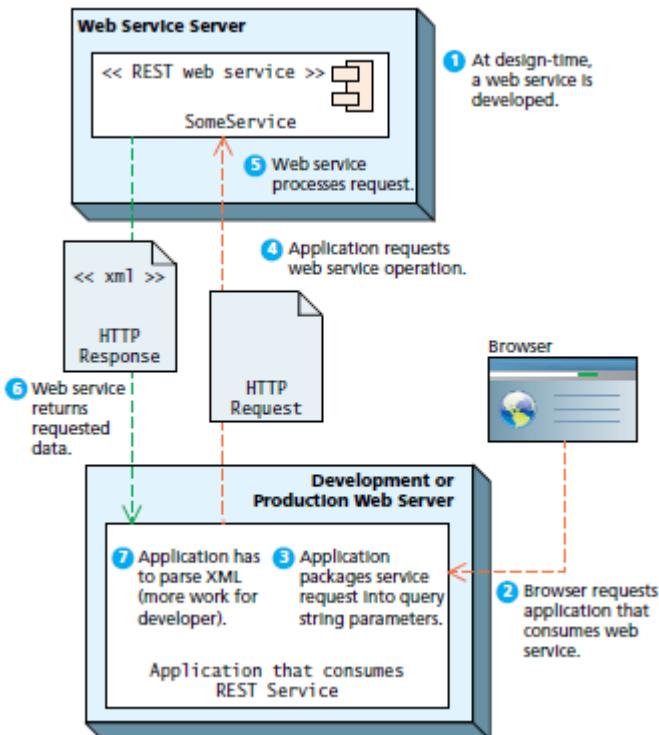
	<pre> <form id="form1" runat="server"> <h1>Web Storage Reader</h1> <script language="javascript" type="text/javascript"> if (typeof (localStorage) === "undefined" typeof (sessionStorage) === "undefined") { alert("Web Storage is not supported on this browser..."); } else { var today = sessionStorage.getItem("TodaysDate"); var artist = sessionStorage.FavoriteArtist; var user = localStorage.UserName; document.write("date saved=" + today); document.write("
favorite artist=" + artist); document.write("
user name = " + user); } </script> </form> </pre>		
--	---	--	--

3	<p>Illustrate various jQuery Selectors with examples.</p> <p>Selectors offer the developer a way of accessing and modifying a DOM object from an HTML page in a simple way. Although the advanced querySelector() methods allow selection of DOM elements based on CSS selectors, it is only implemented in newest browsers. To address this issue jQuery introduces its own way to select an element, which under the hood supports a myriad of older browsers for you! jQuery builds on the CSS selectors and adds its own to let you access elements as you would in CSS or using new shortcut methods.</p> <p>Basic Selectors</p> <p>The four basic selectors are the universal selector, class selectors, id selectors, and elements selectors.</p> <ul style="list-style-type: none"> ■ \$(*) Universal selector matches all elements (and is slow). ■ \$(tag) Element selector matches all elements with the given element name. ■ \$(".class") Class selector matches all elements with the given CSS class. ■ \$("#id") Id selector matches all elements with a given HTML id attribute. <p>For example, to select the single <div>element with id="grab" you would write: <code>varsingleElement = \$("#grab");</code></p> <p>To get a set of all the <a>elements the selector would be: <code>varallAs = \$("a");</code></p> <p>Attribute Selector</p> <p>An attribute selector provides a way to select elements by either the presence of an element attribute or by the value of an attribute. <code>varartistImages = \$("img[src^='/artist/]");</code></p> <p>Pseudo-Element Selector</p> <p>Pseudo-elements are special elements, which are special cases of regular ones. <code>varvisitedLinks = \$("a.visited");</code></p> <p>Contextual Selector</p> <p>These selectors allowed you to specify elements with certain relationships to one another in your CSS. These relationships included descendant (space), child (>), adjacent sibling (+), and general sibling (~).</p> <p>To select all <p>elements inside of <div>elements you would write <code>var para = \$("div p");</code></p> <p>Content Filters</p> <p>The content filter is the only jQuery selector that allows you to append filters to all of the selectors you've used thus far and match a particular pattern. <code>varallWarningText = \$("body *:contains('warning')");</code></p>	6 Marks	(CO2) (PO1,K3)
---	---	---------	-------------------

OR

4	<p>Analyze AJAX requests via jqXHR object.</p> <p>jQuery provides a family of methods to make asynchronous requests. We will start with the simplest GET requests, and work our way up to the more complex usage of AJAX where all variety of control can be exerted.</p> <p>Consider for instance the very simple server time page described above. If the URL currentTime.php returns a single string and you want to load that value asynchronously into the <div id="timeDiv">element, you could write: <code>\$("#timeDiv").load("currentTime.php");</code></p>	6 Marks	(CO3) (PO2,K4)
---	--	---------	-------------------

	<p>GET Requests via the jqXHR Object</p> <pre>var jqxhr = \$.get("/vote.php?option=C"); jqxhr.done(function(data) { console.log(data); }); jqxhr.fail(function(jqXHR) { console.log("Error: "+jqXHR.status); }); jqxhr.always(function() { console.log("all done"); });</pre> <p>POST Requests via the jqXHR Object</p> <pre>var jqxhr = \$.post("/vote.php", "option=C");</pre>	
5	<p>Analyze SOAP services and REST services.</p> <p>SOAP services</p> <p>In the first iteration of web services fever, the attention was on a series of related XML vocabularies: WSDL, SOAP, and the so-called WS-protocol stack (WS-Security, WS-Addressing, etc.). In this model, WSDL is used to describe the operations and data types provided by the service. SOAP is the message protocol used to encode the service invocations and their return values via XML within the HTTP header, as can be seen in Figure.</p> <p>The diagram shows the flow of a SOAP request from a browser to a production web server, which then interacts with a web service server and a development machine. Key components include:</p> <ul style="list-style-type: none"> Web Service Server: Contains a <code><< SOAP web service >></code> box labeled <code>SomeService</code>, a <code><< xml >></code> box, and a <code>WSDL</code> file. Development Machine: Contains a <code>WSDL-Enabled Development Tool</code> and an <code>Application that consumes SOAP Service</code>. Production Web Server: Contains an <code>Application that consumes SOAP Service</code>. Browser: Initiates a <code>SOAP HTTP Request</code> (labeled 6). Sequence of Events: <ol style="list-style-type: none"> At design-time, a web service is developed (tool generally used to generate SOAP processing code, thus less work for developer). (1) At design-time, a WSDL file is generated by tool that describes service. (2) At design-time, tool reads WSDL file to discover the service's operations (methods). (3) At design-time, the tool generates code for consuming service, thus less work for developer. (4) When application is done, it is deployed at design-time onto production web server. (5) Browser requests application that consumes web service. (6) Application requests web service operation. (7) Web service processes request. (8) Web service returns requested data. (9) Tool-generated code parses SOAP-based XML (less work for developer). (10) 	6 Marks (CO3) (PO2,K4)



OR

Demonstrate XML processing in PHP through XMLReader and SimpleXML techniques.

```
// create and open the reader
$reader = new XMLReader();
$reader->open($filename);

// loop through the XML file
while($reader->read()) {
    $nodeName = $reader->name;
    if ($reader->nodeType == XMLREADER::ELEMENT
        && $nodeName == 'painting') {
        // create a SimpleXML object from the current painting node
        $doc = new DOMDocument('1.0', 'UTF-8');
        $painting = simplexml_import_dom($doc->importNode(
            $reader->expand(), true));
        // now have a single painting as an object so can output it
        echo '<h2>' . $painting->title . '</h2>';
        echo '<p>By ' . $painting->artist->name . '</p>';
    }
}
```

6 Marks (CO2) (PO1,K3)

Part B

Imagine you have to animate an element using jQuery. Develop a simple contact form/script that builds and shows a clickable email link when you click the email icon. Also, discuss about the usage of animate() with a step function to do CSS3 rotation.

```
<div class="contact">
    <p>Randy Connolly</p>
    <div class="email">Show email</div>
</div>
<div class="contact">
    <p>Ricardo Hoar</p>
    <div class="email">Show email</div>
</div>
<script type='text/javascript'>
$(".email").click(function() {
    // Build email from 1st letter of first name + lastname
    // @ stroyal.ca
    var fullName = $(this).prev().html();
    var firstName = fullName.split(" ")[0];
```

6 Marks (CO4) (PO3,K5)

```

var address = firstName.charAt(0) + fullName.split(" ")[1] +
    "@mtroyal.ca";

$(this).hide(); // hide the clicked icon.
$(this).html("<a href='mailto:"+address+"'>Mail Us</a>");
$(this).show(1000); // slowly show the email address.
});
</script>

```

Use of animate() with a step function to do CSS3 rotation

```

$(this).animate(
    // parameter one: Plain Object with CSS options.

    {opacity:"show", "fontSize":"120%", "marginRight":"100px"}, 
    // parameter 2: Plain Object with other options including a
    // step function
    {step: function(now, fx) {
        // if the method was called for the margin property
        if (fx.prop=="marginRight") {
            var angle=(now/100)*360; //percentage of a full circle
            // Multiple rotation methods to work in multiple browsers
            $(this).css("transform","rotate("+angle+"deg)");
            $(this).css("-webkit-transform","rotate("+angle+"deg)");
            $(this).css("-ms-transform","rotate("+angle+"deg)");
        }
    },
    duration:5000, "easing":"linear"
});

```

Analyze the document that was shared (<http://backbonejs.org>). Based on this case study, Design a Backbone.js Collection defined to hold Photo Albums and build a custom a view object for the same.

Demonstration of a Backbone.js Collection defined to hold PhotoAlbums

```

// Create a collection of albums
var AlbumList = Backbone.Collection.extend({

    // Set the model type for objects in this Collection
    model: TravelAlbum,

    // Return an array only with the published albums
    GetChecked: function(){
        return this.where({checked:true});
    }
});

// Prefill the collection with some albums.
var albums = new AlbumList([
    new TravelAlbum({ title: 'Banff, Canada', photoCount: 42}),
    new TravelAlbum({ title: 'Santorini, Greece', photoCount: 102}),
]);

```

Deriving custom View objects for our model and Collection

8

6 Marks (CO5)
(PO4,K5)

```

var TravelAlbumView = Backbone.View.extend({
  tagName: 'li',

  events: {
    'click': 'toggleAlbum'
  },

  initialize: function(){
    // Set up event listeners attached to change
    this.listenTo(this.model, 'change', this.render);
  },

  render: function(){
    // Create the HTML
    this.$el.html('<input type="checkbox" value="1" name="" ' +
      this.model.get('title') + '" />' +
      this.model.get('title') + '<span>' +
      this.model.get('photoCount') + ' images</span>');
    this.$('input').prop('checked', this.model.get('checked'));

    // Returning the object is a good practice
    return this;
  },

  toggleAlbum: function() {
    this.model.toggle();
  }
});

```

Course Outcomes (COs)

Students will be able to

CO1	<i>Discuss HTML, CSS, JS, PHP and their semantics to build web pages.</i>
CO2	<i>Apply the underlying web technologies to meet the end user's demands.</i>
CO3	<i>Analyze the static web contents and dynamic web contents of world wide web.</i>
CO4	<i>Design Client-Side Scripts and Server-Side Scripts to generate and display the contents dynamically.</i>
CO5	<i>Investigate a case study which will be given and relate to the underlying core concepts.</i>
CO6	<i>Develop useful applications illustrating the concepts learnt.</i>

Bloom's Taxonomy

K1- Remembering, K2 - Understanding, K3 - Applying, K4 - Analyzing, K5 - Evaluating, K6 - Creating

<i>Course Coordinator</i>	<i>Module Coordinator</i>	<i>Programme Coordinator</i>	<i>HoD</i>



BMS INSTITUTE OF TECHNOLOGY AND MANAGEMENT
YELAHANKA - BANGALORE - 64

Department of Computer Science & Engineering

FOURTH INTERNAL EXAMINATION, NOVEMBER 2018 (CBCS)

Subject: Web Technology and its Applications	Subject Code: 15CS71	Semester : VII CSE 'A'&'B'
Max. Marks : 30	Date: 27- 11 - 2018	Time: 09:30 AM to 11:00 AM

Answer FIVE full questions choosing 3 questions from Part A. Part B is compulsory.

Part A

1	Demonstrate table borders styling along with hover classes with examples.	6 Marks	(CO2) (PO1,K2)
---	---	------------	-------------------

OR

2	Explain the types of using JavaScript with examples.	6 Marks	(CO1) (K2)
---	--	------------	---------------

3	Explain Overlaying and Hidden Elements via CSS.	6 Marks	(CO1) (K2)
---	---	------------	---------------

OR

4	Explain Inline Event Handler approach & Listener approach in JavaScript.	6 Marks	(CO1) (K2)
---	--	------------	---------------

5	How do you include Files in PHP? Illustrate with examples.	6 Marks	(CO2) (PO1,K2)
---	--	------------	-------------------

OR

6	Infer on Mouse Events and Form Events in JavaScript with examples.	6 Marks	(CO3) (PO2,K4)
---	--	------------	-------------------

Part B

7	Imagine you have to design a birthday chooser application where you have to collect customer's birthday, favorite color etc. Develop this application by making use of HTML5's date time controls.	6 Marks	(CO4) (PO3,K5)
---	--	------------	-------------------

8	Based on the case study discussed, Analyze what server we use in our Web Lab and discuss its responsibilities. Also comment on PHP internals.	6 Marks	(CO5) (PO4,K5)
---	---	------------	-------------------

Course Outcomes (COs)

Students will be able to

CO1 Discuss HTML, CSS, JS, PHP and their semantics to build web pages.

CO2 Apply the underlying web technologies to meet the end user's demands.

CO3 Analyze the static web contents and dynamic web contents of world wide web.

CO4 Design Client-Side Scripts and Server-Side Scripts to generate and display the contents dynamically.

CO5 Investigate a case study which will be given and relate to the underlying core concepts.

CO6 Develop useful applications illustrating the concepts learnt.

Bloom's Taxonomy

K1- Remembering, K2 - Understanding, K3 - Applying, K4 - Analyzing, K5 - Evaluating, K6 - Creating

Course Coordinator	Module Coordinator	Programme Coordinator	HoD

BMS INSTITUTE OF TECHNOLOGY AND MANAGEMENT



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

A Brief Report on Student Centric Activity

Name of the Activity/Topic	PBL Activity – WEB – 15CS71
Class/Semester	CSE/7 th semester
Activity Coordinator	Mr. Shankar R
Relevance of the topic	WEB Application Development
Date and Time	4.11.2018 – 9:30-4:30
Venue	CSE Lab 1
Description (Separate Annexure may be enclosed, if the description is exceeding) Photo to be attached separately	Project Based Learning (PBL)-is a student-centered pedagogy that involves a dynamic classroom approach in which it is believed that students acquire a deeper knowledge through active exploration of real-world challenges and problems. Students learn about a subject by working for an extended period of time to investigate and respond to a complex question, challenge, or problem. It is a style of active learning and inquiry-based learning. Gap identified based on CO-PO mapping <ul style="list-style-type: none">• Gaps are identified from PO5 to PO12• PBL projects to be executed by students in teams by using modern tools and a report has to be presented on the same.
No. of students attended	104
Learning outcome	Students got more exposure towards the web application development via modern tool usage, team involvement, project maintenance & lifelong learning, also all the mini projects were evaluated for 5 marks - contributing to direct attainment.
POs achieved/mapped	PO4, PO5, PO9, PO10, PO11, PO12 and PSO1, PSO2.
Total Expenditure in Rs.	Nil

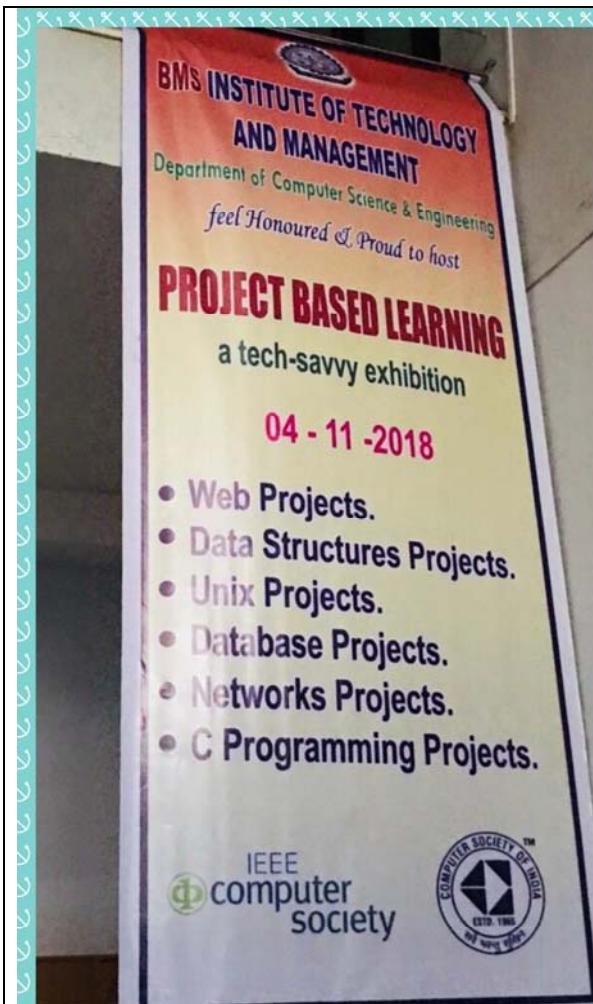
Faculty in-charge

HOD

Encl: Snaps of the event.

SNAPS:





BMS INSTITUTE OF TECHNOLOGY AND MANAGEMENT

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Feedback on PBL Activity - 04/11/2018 - Organized by Mr. Shankar R, Assistant Professor, CSE, BMSIT&M.

Subject: Web Technology and its Applications Code: 15CS71 POs Gaps: PO4, PO5, PO9, PO10, PO11, PO12 and PSO1, PSO2.

Timestamp	Sl. No	Name of the Activity	Name of the Student	USN	E-mail	Mobile No	Sem	Date of conduction	PO4. Can you use the information collected from this session to validate necessary data required for various experiment design?	PO5. Did this session help in understanding the applicability of various modern engineering tools?	PO9. Could the session deliver on significance of working as an individual, member or leader of a multidisciplinary team?	PO10. How useful was the session for communication, report writing and presentation skills?	PO11. How useful was the session for communication, report writing and presentation skills?	PO12. In your opinion has this session contributed in life long learning?	PSO1. Could you Apply design and development principles in the construction of software systems of varying complexity.	PSO2. Could you Analyze the problem and identify computing requirements appropriate to its solution.	Overall feedback of the session: comments, suggestions, remarks.
11/06/18 14:36:21	1	PBL	Abhishek	1BY15CS004	abhishek.sharma1597@gmail.com	8892296369	7	04-11-2018	3	3	2	3	3	3	3	3	It was helpful
11/06/18 14:39:21	2	PBL	Kumar Gaurav Raj	1BY15CS041	kr.gaurav.raj32@gmail.com	886107009	7	04-11-2018	3	3	2	3	3	3	3	3	GOOD,helpful
11/06/18 14:41:21	3	PBL	Godi Lehya Reddy	1by15cs027	lehyareddygod@gmail.com	9100010109	7	04-11-2018	3	2	3	3	2	2	3	3	It was average.
11/06/18 14:43:21	4	PBL	Bhavya Sah	1by15cs018	bhavya751@gmail.com	8296843041	7	04-11-2018	3	3	3	2	3	3	3	3	Great
11/06/18 14:48:21	5	PBL	Kumar Rohit	1by15cs042	rohitranchi@gmail.com	9591824016	7	04-11-2018	3	3	2	1	1	3	3	3	Very good session, 3ly informative
11/06/18 14:54:21	6	PBL	Abhishek b	1BY15CS005	abhishekshaskar27@gmail.com	9740201019	7	04-11-2018	3	3	2	2	3	3	3	3	It was very enlightening session
11/06/18 15:00:21	7	PBL	Deeptha S	1BY15CS023	deepthas13@gmail.com	9482253490	7	04-11-2018	3	3	3	3	3	3	3	3	it is most important
11/06/18 15:05:21	8	PBL	Ketan Yadav	1by15cs041	ketan8085@gmail.com	8189861906	7	04-11-2018	3	2	3	3	2	2	3	3	Good
11/06/18 15:08:21	9	PBL	Chandra Shekar	1BY15CS020	chandrashekar03@gmail.com	7022894658	7	04-11-2018	3	2	3	2	3	3	3	3	Superb
11/06/18 15:11:21	10	PBL	Adarsh Kumar Sah	1BY15CS006	adarshkumar.sah6@gmail.com	7411929815	7	04-11-2018	3	3	2	3	3	3	3	3	Good
11/06/18 15:14:21	11	PBL	Harsit Choudhary	1BY15CS029	harsit12345c@gmail.com	7760915483	7	04-11-2018	3	2	3	2	1	3	3	3	It was average.
11/06/18 15:19:21	12	PBL	Harsit Choudhary	1BY15CS029	harsit12345c@gmail.com	7760915483	7	04-11-2018	3	2	3	2	2	2	3	3	I'm not much confident
11/06/18 15:23:21	13	PBL	Amogh G	1BY15CS008	amoghsk279@gmail.com	9742454257	7	04-11-2018	3	2	3	2	2	2	3	3	Very good session, 3ly informative
11/06/18 15:25:21	14	PBL	Nikhil N Achar	1BY15CS057	nikhilachar97@gmail.com	8123018956	7	04-11-2018	3	2	3	2	3	3	3	3	it is most important
11/06/18 15:32:21	15	PBL	N.PoornimaNischala	1BY15CS059	poornimanischala@gmail.com	9071323104	7	04-11-2018	2	3	2	3	3	3	3	3	it is most important
11/06/18 15:39:21	16	PBL	ARUNDHATI S SHANBHAG	1BY15CS013	arus2396@gmail.com	8951536800	7	04-11-2018	3	3	3	3	2	3	3	3	Good
11/06/18 15:45:21	17	PBL	ADITYA SHIVA	1by15cs113	ksaditya05@gmail.com	9686582526	7	04-11-2018	3	3	3	2	3	3	3	3	It was very helpful and informative
11/06/18 15:50:21	18	PBL	Ruchita Iyer	1BY15CS072	13ruchitaiyer@gmail.com	7259267935	7	04-11-2018	3	1	3	2	2	2	3	3	Good
11/06/18 15:53:21	19	PBL	Tejas Vaidya	1BY15CS093	tejas1998@gmail.com	8095085628	7	04-11-2018	3	3	2	3	3	3	3	3	Superb
11/06/18 15:57:21	20	PBL	Prafulla Sreevalsan	1BY15CS063	prafulla2497@gmail.com	7978742133	7	04-11-2018	3	3	3	3	3	2	3	3	Good
11/06/18 16:03:21	21	PBL	Akshay AB	1BY15CS112	akshaybandarkar2000@gmail.com	9980912003	7	04-11-2018	2	3	2	2	2	2	3	3	Good
11/06/18 16:06:21	22	PBL	Vighnesh Raut	1by15cs100	vighnesh.raut13@gmail.com	9886661305	7	04-11-2018	3	3	2	3	3	3	3	3	Very helpful.
11/06/18 16:12:21	23	PBL	Vinith VB	1BY15CS114	vinithbandarkar2000@gmail.com	9980912004	7	04-11-2018	3	2	3	3	2	3	3	3	Superb
11/06/18 16:16:21	24	PBL	Sinchana MK	1by16cs415	sinchanamk@gmail.com	7760619890	7	04-11-2018	3	1	3	2	2	3	3	3	Good
11/06/18 16:20:21	25	PBL	Thatode sai surya abhishek	1by15cs108	saisurya.abhishek@yahoo.com	8861584667	7	04-11-2018	3	2	3	2	2	2	3	3	Very good session, 3ly informative
11/06/18 16:26:21	26	PBL	MOHAMMED ZAMAN	1BY15CS052	mohammedzaman.mz1@gmail.com	9066787252	7	04-11-2018	3	2	3	2	1	2	3	3	it is most important
11/06/18 16:30:21	27	PBL	Shalini BN	1BY15CS080	bnsalinai28@gmail.com	9901397955	7	04-11-2018	3	3	2	3	3	3	3	3	Good
11/06/18 16:37:21	28	PBL	bhavesh m	1by15cs017	bhaveshkumar110@gmail.com	7676455551	7	04-11-2018	3	3	3	2	3	3	3	3	Superb
11/06/18 16:42:21	29	PBL	MOHEIN RANJITKAR	1BY15CS053	moheinranjikar@gmail.com	9740268603	7	04-11-2018	3	3	3	3	3	2	3	3	Good
11/06/18 16:47:21	30	PBL	Mayank kushal	1BY15CS048	mayankkushal26@gmail.com	7795206707	7	04-11-2018	3	2	3	2	2	2	3	3	It was average.
11/06/18 16:56:21	31	PBL	samar khan	1by15cs077	samar Khan7277@gmail.com	7259474699	7	04-11-2018	3	2	3	2	2	3	3	3	I'm not much confident
11/06/18 16:58:21	32	PBL	Ranjitha R	1BY16CS409	pegglestar@gmail.com	9035341413	7	04-11-2018	3	3	2	2	2	3	3	3	Very good session, 3ly informative
11/06/18 17:00:21	33	PBL	Sourabh bangar	1BY15CS106	bangarsourabh38@gmail.com	8618911060	7	04-11-2018	3	3	3	3	3	3	3	3	It was average.
11/06/18 17:09:21	34	PBL	Manish K	1BY15CS046	manishmanik11@gmail.com	9740504004	7	04-11-2018	3	3	3	2	3	3	3	3	Highly helpful.
11/06/18 17:13:21	35	PBL	HARSHITH T R	1by15cs039	tfrashith147@gmail.com	7899348954	7	04-11-2018	3	3	2	3	3	3	3	3	Great session..
11/06/18 17:21:21	36	PBL	P vignesh	1by15cs061	Vignesh.p6315@gmail.com	9880615924	7	04-11-2018	3	2	3	3	2	2	3	3	It was an interesting session.
11/06/18 17:27:21	37	PBL	Divyashree G	1by15cs025	divgangadhar963@gmail.com	8747078495	7	04-11-2018	3	3	2	3	3	3	3	3	Great
11/06/18 17:33:21	38	PBL	Bhanu pratap singh	1by15cs015	bhanupratap784@gmail.com	7899068126	7	04-11-2018	3	3	2	1	1	3	3	3	Excellent
11/06/18 17:37:21	39	PBL	BHAVYA K SHETH	1BY15CS110	bhavya.sheth5@gmail.com	9611436888	7	04-11-2018	3	3	2	2	3	3	3	3	Very helpful.
11/06/18 17:43:21	40	PBL	Umesh Dattari	1by15cs097	umeshdattani@gmail.com	94437393635	7	04-11-2018	3	3	3	3	3	3	3	3	Awesome
11/06/18 17:46:21	41	PBL	VISHNU MENON	1by15cs102	vizu.m07@gmail.com	8982840823											