Name:

Enrolment No:

UNIVERSITY OF PETROLEUM AND ENERGY STUDIES
End Semester Examination, December 2022

Course: Responsive Mobile Platform	**Semester: V**
Program: BTech CSE (All Branches)	**Time: 03 hrs.**
Course Code: CSMC3019P	**Max. Marks: 100**

Instructions: Attempt all Questions

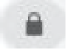
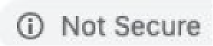
SECTION A
(5Qx4M=20Marks)

S. No.		Marks	CO
Q 1	A website could choose to hand out the same cookie for both HTTP and HTTPS and not have to worry about different cookie values. Explain the advantage for the backend engineer of having the same cookie for both protocols.	**4**	**CO1**
Q 2	Illustrate how DNS can be used to do load balancing across a web application's servers.	**4**	**CO2**

Q 3	For each of the code fragments below state what the console.log message will print. If the execution aborts due to an error, describe the error.
	1. var a = 3;
	console.log(a);
	2. console.log(a);
	var a = 3;
		4	**CO3**

Q 4	If you resize a modern web app in a window, it is not unusual to see it just shrink to fit in the smaller window but if you make the window small enough it will re-layout the app to better work in the small window. Explain the browser’s mechanism that web app used to have this behavior.	**4**	**CO4**
Q 5	Illustrate what the browser will do when it finds this line in an HTML document: <script type=”text/javascript” src=”/main.js”></script>	**4**	**CO1**

SECTION B
(4Qx10M= 40 Marks)

Q 6	If you display a web page in the Chrome browser that is entirely fetched using the HTTPS protocol, Chrome will display  next to the URL. If you change the web page to include a small image of something public (e.g. a logo) and use HTTP to fetch the image Chrome will display  ⓘ Not Secure next to the URL as if you didn't use HTTPS for anything. Explain why the Chrome developers could justify slapping "Not Secure" on something seemingly benign as an HTTP fetch of a small image. Describe the attack they are worried about.	**10**	**CO2**
Q 7	Web app-based discussion forms frequently define their own markup language rather than simply using HTML and having the browser's rendering engine do the markup. Besides syntax complexity disadvantage of HTML over something like Markdown, describe the key security problem with using HTML as a markup language in a web application.	**10**	**CO3**
	OR		
Q 7	If you receive an email and click on a link for https://www.bankofthevvest.com (note: vvest, not west.) Assume that www.bankofthevvest.com is under the attacker's complete control. Will your browser provide indication that the site you are visiting is not legitimate? If not, explain why. If so, how would it likely show up?	**10**	**CO3**
Q 8	Consider the HTML snippet below: <div id="mainDiv"><p>text 1a<span>text 1b</span></p><p>text 2a<span>text 2b</span></p></div> Sketch out the DOM tree representation of this HTML snippet, showing only the parent/child relationships between the code. Non-text nodes in the tree should be labeled with their tag names, whereas text nodes should be labeled with their value.	**10**	**CO4**
Q 9	Imagine you have a set of web applications running in a browser. Some of the web applications are considered to be single-page applications, while others are not. Describe how you could differentiate between the two types by examining their behavior in the browser	**10**	**CO1**
SECTION-C **(2Qx20M=40 Marks)**			

Q 10	The Node.js runtime includes a function setImmediate that allows the call to directly add a callback to the Node.js event loop. It is like the DOM's setTimeout function except there is no delay.
	Consider the following Node.js program:
	for (var i = 0; i < 2; i++) {
	setImmediate(function(err) {
	console.log(i);
	});
	console.log(4);
	}
	a) Write down what is printed after the for-block code above is executed.
	Suppose the setImmediate call is wrapped in an immediately invoked function expression like:
	for (var i = 0; i < 2; i++) {
	(function (i) {
	setImmediate(function(err) {
	console.log(i);
	});
	})(i);
	console.log(4);
	}
	b) Does this affect the order/contents of what is printed to the console? If yes, state so and write down what is now printed after the for block above is executed (Order matters). If no, state so and briefly justify your answer.
		20	**CO2**

Q 11	Consider the code fragment below. Suppose that there are a total of 5 users in the database. Given the endpoint below, the request to "/user/countusers" is supposed to return the number of users in the system but has a required line commented out.
	var User = require('./schema/user.js');
	var async = require('async');
	app.get("/user/countusers", function(request, response) {
	var count = 0;
	User.find({}, function (err, users) {
	if (err) {
	response.status(400).send(JSON.stringify(err));
	return;
		20	**CO3**

	}
	//(A) response.status(200).send(JSON.stringify(count));
	async.each(users, function(user, callback) {
	count += 1;
	//(B) response.status(200).send(JSON.stringify(count));
	callback();
	}, function(err) {
	//(C) response.status(200).send(JSON.stringify(count));
	});
	})
	//(D) response.status(200).send(JSON.stringify(count));
	});
	
	a. There are four commented-out Express response-sending statements labeled (A)-(D). The statements either return the correct count, return an incorrect count but otherwise run successfully, or will use Express incorrectly and likely generate an error. For each of the statements say which of the above it will be if the line is uncommented. If the code isn't using Express incorrectly, state what value for the number of users is returned.
	
	b. If the code is run as is, all the response lines commented out, describe the symptoms you would see in a web app attempting to use the endpoint.		

	OR		

Q 11	When defining routes in Express.js we use a call of the form:
	app.get(urlPath, callbackFunction);
	so that the callbackFunction will be invoked when an HTTP GET request with a path of urlPath comes in.
	
	The number of arguments in the definition of the callbackFunction varies. Some route definitions have three arguments:
	app.get(urlPath, function (request, response, next) { …
	while others only use two arguments:
	app.get(urlPath, function (request, response) { …
	
	1. When should the next parameter be used by the callback?
	
	2. Would it make sense to have callback with only the request parameter:
	app.get(urlPath, function (request) { …
	Justify your answer.
		20	**CO3**