Course Assessment Test

Course Title	Backend Web Development with Node.js	Date	Oct 2025
Name		Dept	

1) *This assessment test is to be given out before course commencement. Answers are to be filled in column entitled "Pre-Course Answer"*
2) *At the end of the course, the same assessment sheet is to be given out where answers are to be filled in column entitled "Post-Course Answer". Instructor will then share the answers and participants need to total the score in both "Pre" and "Post" columns through self-marking.*
3) *Assessment sheets will be collected for filling.*

No	Question	Pre-Course Answer	Post-Course Answer
1	Which of the following statements most correctly characterizes the relationship between Node.js and Express.js? a) Express is a cross-platform runtime environment for JavaScript code while Node.js is a web application framework within the Express ecosystem of packages / libraries b) Node.js is a cross-platform runtime environment for JavaScript code while Express.js is a web application framework within the Node ecosystem of packages / libraries c) Both Express and Node are alternative cross-platform runtime environments for JavaScript d) Both Express and Node are alternative JavaScript web application frameworks		B
2	What is the advantage of the single thread event driven architecture of Node that uses an event loop? a) Allows it to handle large number of concurrent connections efficiently making it easily scalable for microservices architecture b) Optimizes memory usage as one thread is ever only running at one time c) Allows more effective context switching when a new thread assumes priority as the single thread d) Supports more effective CPU utilization on modern multi core CPU architecture		A
3	What is the key feature of Express architecture that allows it to perform various functionalities such as processing incoming requests step by step (e.g., authentication, logging, parsing JSON, etc)? a) Asynchronous non-block event driven model b) Layered MVVC c) Middleware d) Routing		C

4	Which of the following HTTP Request methods is used primarily to update an existing resource on a backend service referenced by an ID / URI? a) GET b) POST c) DELETE d) PUT		D
5	What does HTTP status code 404 in a return response from a backend server mean? a) client is not authenticated yet to the server b) client is not authorized to retrieve requested resource from the server c) the requested resource cannot be located on the server d) error occurred on the server that cannot be handled causing it to crash		C
6	Which of the following are the core features of microservices? i. Decentralized Data Management - Each microservice manages its own database. ii. Independently Deployable - Teams can develop and deploy services independently. iii. Scalability - Services can scale independently depending on the load of each service iv. Technology Agnostic - Different services can use different languages and frameworks a) Feature i), ii) and iii) b) Feature i), iii) and iv) c) Feature ii), iii) and iv) d) All features		D
7	What is the key feature of Docker containers that make it so ideal for deploying microservices architecture? a) Containers provide standardization and portability of application environment configuration b) Containers provide ability to fine tune host resource consumption from within the applications that are running within the container itself c) Containers provide advanced network facilities that speed up networking between apps as compared to running the apps directly on the host server infra itself d) Containers share a common root file system between each other on the same host machine, permitting easy synchronization of file content between multiple containers		A

8	What is the purpose of this statement that is typically placed at the end of an Express app?
	
	`app.use((req, res) => {`
	`  res.status(404).send('Error in app !')`
	`})`
			B

	a) It is an error handler that sends a 404 error response in the event that the HTML file requested from the app does not exist in the project directory holding app.js b) It is catch-all for non-matching routes in HTTP requests sent to the app using the middleware architecture of Node.js c) It provides an option for error handling at the later portion of the script in case error handling code is not properly implemented earlier d) It allows the specification of the status code of the error message to be returned to the user in the event an error occurs in the parsing of the content of the POST message sent to the app		

9	Consider an Express app with the following route definitions in its body that has just been started up at port 3000 on the local machine:
	`....`
	`....`
	`....`
	`app.get('/tryme', (req, res) => {`
	`  res.send('1st GET request received')`
	`})`
	`app.post('/tryme', (req, res) => {`
	`  res.send('POST request received')`
	`})`
	`app.get('/tryme', (req, res) => {`
	`  res.send('2nd GET request received')`
	`})`
	`.....`
	If this URL `localhost:3000/tryme` was entered into a browser, what would be the result?
	a) The app would crash
	b) The response `1st GET request received` is returned
	c) The response `2nd GET request received` is returned
	d) The response POST request received is returned
			B

10	Which of the following code snippets below correctly extract query parameters from a URL and return them in the response to the client?
	a)
	`app.get('/search/:first/:second', (req, res) => {`
	`  const { first, second } = req.query;`
	`  res.send(\`Retrieved parameters : ${first} and ${second}\`);`
	`});`
			C

b)

```
app.get('/search', (req, res) => {
  res.send(`Retrieved parameters : ${req.params.first} and
${req.params.second}`);
});
```

c)

```
app.get('/search', (req, res) => {
  const { first, second } = req.query;
  res.send(`Retrieved parameters : ${first} and ${second}`);
});
```

d)

```
app.get('/search', (req, res) => {
  res.send(`Retrieved parameters : ${query.first} and
${query.second}`);
});
```

Total			