

**COMSATS** **University Islamabad, Lahore Campus**



**Terminal Examination – Fall 2021**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Course Title: | Web Technologies | | | | Course Code: | | CSC336 | Credit Hours: | 3(2,1) |
| Course Instructor/s: | Shuja Akbar, Muhammad Ali Khan | | | | Program Name: | | BCS | | |
| Semester: | 6th | Batch: | SP19 | Section: |  | | Date: | 05/01/2022 | |
| **Time Allowed:** | **3 hours** | | | | **Maximum Marks:** | | | **50** | |
| Student’s Name: |  | | | | Reg. No. | CIIT/SDP- /LHR | | | |
| **Important Instructions / Guidelines:**   1. Assume the use of React JS, Node JS, MongoDB, and Express JS | | | | | | | | | |

**Question 1: [Marks: 6]**

**Why is routing handled at both the frontend and the backend in MERN-based applications? Provide one example to justify your answer.**

Routing is handled both at the frontend and the backend in MERN-based applications to provide a seamless user experience, better performance, and to implement features like server-side rendering, data validation, and access control………………………………….

**Question 2: [Marks: 5]**

**Which JavaScript request handling approach is used at the backend to deal with performing operations on MongoDB? Provide one example to justify your answer.**

In the ReactJS and Node.js ecosystem, the most commonly used JavaScript request handling approach to deal with performing operations on MongoDB is to use an Express.js server…………………….

**Question 3: [Marks: 6]**

**Write two custom middleware’s called “authenticate” and “authorize”. The authenticate middleware passes control to the authorize middleware.**

**Question 4: [Marks: 12]**

**Line 1 in exam.js returns an error because it is a blocking code. Implement exam.js using callback, promise, and async and await calls to remove the error/s. The code returns cgpa as an object with value cgpa.**

**Graphical user interface, text, application

Description automatically generated**

**Question 5: [Marks: 6]**

**Assume the use of MongoDB, create collection “animals” to hold document with the following key value pairs- name of type String, legs of type Number, and availability of type Boolean. Write complete code including the use of any package, if required.**

**Question 6: [Marks: 6]**

**HTTP protocol helps clients interact with the server. Describe any two HTTP request methods used to transfer data from client to server. Also, discuss any two methods used to perform operations on MongoDB at the backend.**

**Question 7: [Marks: 9]**

1. **Discuss the change in at least three files/folders when a third-party package is installed in a MERN stack application.**

* Package Json
* Package lock Json
* Node modules folder

1. **What is the relation between Package.json and Package\_lock.json files?**

The `package.json` and `package-lock.json` files work together to manage project dependencies in Node.js projects.

1. Dependency Information:

- The `package.json` file contains metadata about the Node.js project, including the list of dependencies required for the project to run. ‘

- `package-lock.json`: The `package-lock.json` file is automatically generated by npm when installing or updating dependencies. It includes detailed information about the exact versions of the dependencies installed.

2. Deterministic Dependency Management:

- The `dependencies` field in `package.json` specifies the desired version range for each dependency.

- The `package-lock.json` file locks the exact versions of the dependencies installed during the initial installation.

3. Reproducible Builds:

- the `package.json` file defines the range of versions allowed for each dependency, it does not guarantee that all developers or environments will get the same versions during installation. Without `package-lock.json`, different installations could potentially lead to different versions of the dependencies, causing inconsistencies and potential issues.

4. Version Consistency

- Developers can update the `package.json` file manually to add or remove dependencies, specify new version ranges, or update other project settings.

- The `package-lock.json` file remains unchanged unless new dependencies are installed or the dependencies are explicitly updated using npm commands.

1. **Why Express JS is useful to handle backend operations instead of core Node JS?**

Express.js is a web application framework built on top of Node.js that simplifies the process of handling backend operations. While Node.js provides the core functionality for building server-side applications, Express.js enhances and extends Node.js by offering several benefits that make it more useful for backend development:

* Routing and Middleware:
* Simpler Code Structure:
* Wide Community Support:
* Middleware Ecosystem:
* Flexibility:
* Error Handling:
* HTTP Utility Methods: