

STUDENT WORK

NODE.JS, EXPRESS.JS, MYSQL SERVER AND DATABASE CONNECTION

Topics: Node.js • NPM • Express.js • Routes • MySQL • mysql2 • Database Connection • SELECT Query • REST API

8. ONLINE-ONLY WORK AND SUBMISSION

Students must complete and submit this work online.

Do NOT copy, paste, cut, take screenshots/screen captures, use a phone to photograph the work or computer screen, or print the work for submission. Submit the work through the online submission platform in the required digital format.

GENERAL INSTRUCTIONS

1. Answer all questions.
2. Section A contains multiple-choice and fill-in-the-blank questions.
3. Section B requires you to write complete working programs.
4. Use Node.js and Express.js syntax correctly.
5. Use MySQL2 for database connectivity.
6. Test your programs and correct any errors before submission.
7. Do not copy programs without understanding how they work.
8. **CREATE YOUR OWN WORD DOCUMENT AND INSERT THOSE SCREESSHOTS AND PRINT OUT (SO AS TO BE SUBMITTED TOGETHER WITH THE HAND WRITTEN DOCUMENTS).**

SECTION A: KNOWLEDGE AND UNDERSTANDING

PART I: MULTIPLE-CHOICE QUESTIONS

20 Questions × 1 Mark = 20 Marks

Choose the **best answer**.

1. What is Node.js?

- A. A database management system
- B. A JavaScript runtime environment

- C. A web browser
- D. A programming language

2. What does NPM stand for?

- A. Node Programming Method
- B. New Package Manager
- C. Node Package Manager
- D. Network Package Module

3. Which command initializes a Node.js project with default settings?

- A. `node start`
- B. `npm init -y`
- C. `npm create node`
- D. `node init project`

4. Which command installs Express.js?

- A. `npm install express`
- B. `node install express`
- C. `npm express create`
- D. `express install node`

5. Express.js is mainly used to:

- A. Create databases
- B. Build web servers and backend applications
- C. Design images
- D. Manage computer hardware

6. Which code correctly imports Express using CommonJS?

- A. `import express from "express";`
- B. `include express;`
- C. `const express = require("express");`
- D. `using express;`

7. Which function creates an Express application?

- A. `express.server()`
- B. `express()`
- C. `create.express()`
- D. `new.server()`

8. Which HTTP method is normally used to retrieve data?

- A. POST
- B. PUT
- C. DELETE
- D. GET

9. What does `req` represent in an Express route?

- A. Database
- B. Request
- C. Response
- D. Server port

10. What does `res` represent?

- A. Response
- B. Request
- C. Route
- D. Database

11. Which port is the common default port for MySQL?

- A. 80
- B. 3000
- C. 3306
- D. 8080

12. Which Node.js package is used in the lesson to connect Node.js with MySQL?

- A. `express-db`
- B. `mysql2`
- C. `databasejs`
- D. `sqlnode`

13. Which command installs `mysql2`?

- A. `npm install mysql2`
- B. `node install mysql2`
- C. `npm mysql2 install`
- D. `mysql install node`

14. Which SQL command retrieves records from a table?

- A. INSERT
- B. UPDATE
- C. SELECT
- D. DELETE

15. What does the following query do?

```
SELECT * FROM students;
```

- A. Deletes all students
- B. Displays all columns and records from students
- C. Creates the students table
- D. Updates all students

16. Which code creates a MySQL connection?

- A. `mysql.connectDatabase()`
- B. `mysql.createConnection()`
- C. `mysql.openServer()`
- D. `mysql.database()`

17. Which database name is used in the practical example?

- A. university
- B. college
- C. school
- D. studentdb

18. Which Express code defines a GET route?

- A. `app.routeGet()`
- B. `app.get()`
- C. `app.fetch()`
- D. `app.request()`

19. Which command starts a Node.js file named `server.js`?

- A. `npm server.js`
- B. `start server.js`
- C. `node server.js`
- D. `run node server`

20. Which URL can be used to test an Express server running on port 3000?

- A. `http://localhost:3306`
 - B. `http://localhost:3000`
 - C. `http://mysql:3000`
 - D. `http://server:80`
-

PART II: COMPLETE THE BLANK SPACES

10 Questions × 1 Mark = 10 Marks

Fill in each blank with the correct word, command, method, or concept.

21.

Node.js allows JavaScript to run outside a _____.

22.

NPM stands for Node _____ Manager.

23.

The command used to initialize a Node.js project is:

24.

The command used to install Express.js is:

25.

The command used to install the MySQL2 library is:

26.

The default MySQL port used in the practical example is _____.

27.

The MySQL database used in the example is called _____.

28.

The SQL command used to retrieve data from a table is _____.

29.

In Express.js, `req` represents the _____ sent by the client.

30.

In Express.js, `res` represents the _____ sent back by the server.

SECTION B: PRACTICAL CODING/20 Marks

Write complete programs. Your programs must be properly formatted and should be tested.

QUESTION 31: CREATE THE MYSQL DATABASE

4 Marks

Create a MySQL database called:

`school`

Then create a table called:

`students`

The table must contain the following fields:

Field	Data Type	Requirement
id	INT	Primary Key, Auto Increment
name	VARCHAR(100)	NOT NULL
age	INT	—
class	VARCHAR(50)	—

Write the complete SQL program.

QUESTION 32: INSERT STUDENT RECORDS

3 Marks

Insert **at least five students** into the `students` table.

Your records must contain:

- Student name
- Age
- Class

Example format:

```
INSERT INTO students (name, age, class)
VALUES
('Jean', 20, 'L5SOD'),
('Alice', 21, 'L5SOD');
```

Write your own complete INSERT statement containing at least five students.

QUESTION 33: CREATE A NODE.JS + MYSQL CONNECTION

4 Marks

Create a Node.js program called:

`database.js`

Your program must:

1. Import `mysql2`.
2. Create a connection to MySQL.
3. Use:
 - `host: localhost`
 - `user: root`
 - `database: school`
 - `port: 3306`
4. Test the connection.
5. Display a success message when the connection is successful.

6. Display an error message if the connection fails.

Your program should follow the connection approach taught in class. The reference material demonstrates `mysql.createConnection()` and `connection.connect()` with error handling.

QUESTION 34: CREATE AN EXPRESS SERVER

3 Marks

Create a Node.js Express server called:

`server.js`

The server must:

1. Import Express.
2. Create an Express application.
3. Use port 3000.
4. Create a GET route `/`.
5. Display:

Welcome to Student Management System

6. Start the server.

Test it using:

`http://localhost:3000`

QUESTION 35: CONNECT EXPRESS TO MYSQL AND SELECT STUDENTS

6 Marks

Create a complete Express application that connects to the `school` MySQL database and displays all students.

Your program must:

Step 1

Import:

```
express
mysql2
```

Step 2

Create an Express application.

Step 3

Create a MySQL connection or connection pool using:

```
host: localhost
user: root
password: your_password
database: school
port: 3306
```

Step 4

Create the following route:

```
GET /students
```

Step 5

Execute:

```
SELECT * FROM students;
```

Step 6

Return the database results as JSON.

Step 7

Return an appropriate error response if the database query fails.

Expected URL

```
http://localhost:3000/students
```

The course material demonstrates this exact backend flow: Express receives `/students`, `mysql2` executes `SELECT * FROM students`, and the results are returned as JSON.

QUESTION 36: COMPLETE BACKEND PROJECT

INTEGRATED PRACTICAL TASK

Optional Extension / Challenge

Create a complete **School Student Management API**.

Your project should contain:

```
school-api/  
├──  
├── package.json  
├── server.js  
└── database.js
```

Requirements

A. Database

Create:

```
school
```

and:

```
students
```

with at least five student records.

B. Node.js

Initialize the project using:

```
npm init -y
```

C. Install dependencies

Install:

```
npm install express
```

and:

```
npm install mysql2
```

D. Express server

Run the server on:

```
PORT = 3000
```

E. API route

Create:

```
GET /students
```

F. SQL

Use:

```
SELECT * FROM students;
```

G. JSON response

The endpoint should return the student records as JSON.

H. Testing

Test:

```
http://localhost:3000/
```

and:

```
http://localhost:3000/students
```

You may use a browser, Postman, or another HTTP client. The practical exercise in the teaching material similarly asks students to create the database, install Express and mysql2, create the server, connect to MySQL, create `GET /students`, and test the API.

SUBMISSION REQUIREMENTS

Each student must submit:

1. MySQL database creation SQL.
2. Student table creation SQL.

3. Student INSERT statements.
4. `database.js`.
5. `server.js`.
6. (CREATE YOUR OWN WORD DOCUMENT AND INSERT THOSE SCREESHOTS AND PRINT IT(TO BE SUBMITTED TOGETHER WITH THE WRITTEN DOCUMENTS)).
7. Screenshot showing successful MySQL connection.
8. Screenshot showing the Express server running.
9. Screenshot of:

`http://localhost:3000/students`

10. The returned student records must be visible in JSON format.

IMPORTANT

Students should understand the complete flow:

```
CLIENT
  ↓
HTTP REQUEST
  ↓
EXPRESS SERVER
  ↓
NODE.JS
```

6. ONLINE SUBMISSION REQUIREMENTS

6. Submit the completed work online in the required digital format.
7. Do NOT submit screenshots, screen captures, phone screenshots, photographs, or photos of a computer/phone screen.
8. Do NOT print the assignment or submit scanned/photographed printed pages.
9. Do NOT copy, paste, or cut material from another student's work or from prepared answers. Students must produce their own work online.
10. Submit the actual project files/code and other files specifically requested by the online submission platform.