Assignment 3 - 3-Tier application

This assignment has five parts and focuses on the development of a simple posting system based on nodejs & mysql. In this assignment, you will train:

- the development of a stateful client-server application,
- use of asynchronous fetch
- testing of your code

Please provide the Dockerfile and/or docker-compose.yml you used in completing the assignment. Failure to provide the needed Dockerfile and/or docker-compose.yml will make it impossible to test the solution you developed and result in 0 marks for this assignment.

Part A) 20 Points
Develop a get method with the patch /init that creates a database named postdb and a table named posts.

Part B) 20 Points
Develop a post method with the patch /addPost that expects data and topic in the body of a request. This method will insert the topic and data into the table posts.

Part C) 20 Points
Develop a get method with the patch /getPosts that returns all the entries in the table posts.

Part D) 20 Points
Develop a web page that allows the user to create a post (topic & data) and submit it (e.g. pressing a button) via an asynchronous fetch to the server.
Also, provide a mechanism for the user (e.g. pressing a button) to retrieve all existing posts in the database via an asynchronous fetch.

Part E) 20 Points
Write a test report that shows how you tested the nodejs code you developed (e.g. be sure to use loadtest [https://www.npmjs.com/package/loadtest] for the performance test). Your report (short paragraph) should answer the following questions

- How did you test your code?
- How long does it take to process a single post (performance)?
- Does the size of the data submitted to the server impact the performance?
- How does the number of requests impact the performance of the server?
- How does the level of concurrency impact the performance of the server?

What to hand in?

Dockerfile and docker-compose.yml file that you used in the assignment. Failure to provide this/these files will result in a zero (0) grade for the assignment.

Part a&b&c) -> one file called server.js (make sure that calling nodejs server.js will work).
Part d) -> one file called posting.html (this file should contain all JS code and HTML)
Part e) -> one file called report.pdf that contains your report.