



there was something i have had in plan like an app an app where i case of some accident a person ccan open app inform the police ambulance fire fighter etc, and the location is updated there are many circumstances where people feels difficulty explaining the location to the officials, and i also want to integrate an ai system where people can ask the ai what primary step they should take if someone is injured, if there is a fire , a crime etc so they can help , and much more

ChatGPT

This is actually a **very strong idea** and it fits your SDC rubric almost *too* well. It solves a real problem, it naturally needs cloud services, and it gives you room to show both engineering and thinking. Let's shape it into something your sir will immediately recognize as a **serious AWS-based system**, not just an app idea.

I'll treat this like we're designing it together over coffee, not writing mar ↓ ng fluff.



Ask anything





FOR THE TWO NUMBER (S), AND WE CONTINUE.

Saleh Bint E Bilal

first of all, we can't use lambda,  
so i suppose ec2 should be  
used instead and secondly  
what should i start from and  
how

SB

ChatGPT

Perfect, thanks for clarifying 

This actually **does NOT** weaken your project at all. Using **EC2 instead of Lambda is completely acceptable** for your rubric, as long as you justify it properly (which I'll show you).

I'll answer in **two parts**, very clearly:

---

## PART 1: Using EC2 Instead of Lambda (Is it OK?)



**YES, 100% OK**

Your rubric **does not mandate Lambda**. It asks for:

- AWS-based implementation



Ask anything



## SmartSOS: AWS-Based Emergency...

1. Networking layer
2. Compute layer
3. Data layer
4. Security layer
5. Monitoring layer

This makes your diagram professional and easy to understand for your project submission.

givve me mermaid code to draw this aws diagram



Here's the Mermaid code to create a professional AWS architecture diagram for your SmartSOS project:

### Complete Mermaid Code for AWS Diagram:

Diagram

Code

Save

Fullscreen

Message DeepSeek

DeepThink

Search

+

↑



areesha\_hameed\_khan@DESKTOP-QMQ24B3:/mnt/c/Users/Areesha/Desktop/smartsossss/frontend\$ npm r...



The reason your app is still crashing is that your terminal is still executing the **backend** configuration (trying to run server.js with nodemon), even though you are in the frontend folder.

Since you have switched to a "No-DB" (Mock) version, you need to force the project to use frontend tools (Vite) instead of backend tools.

Follow these steps exactly to fix the package.json and start the app:

### 1. Correct the package.json

Open the file at /mnt/c/Users/Areesha/Desktop/smartsossss/frontend/package.json and replace everything with this code. This tells the computer to run the web interface instead of a database server:

json



Ask Gemini



Fast





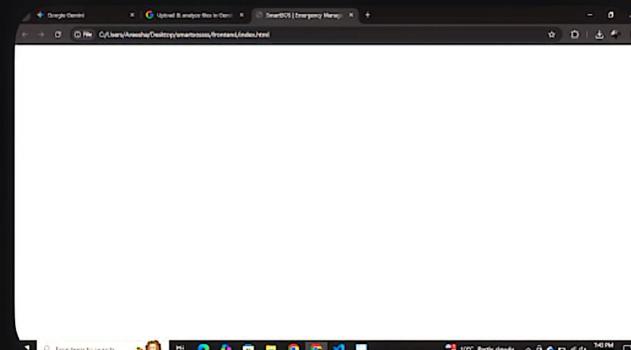
smartsossss



is this correct folder structure everything  
is okay?



The attached zip contains too many files to process.  
For more information, see [here](#).



<!DOCTYPE html> ▾  
<html lang="en">  
<head>...



Analysis ▾



Ask Gemini

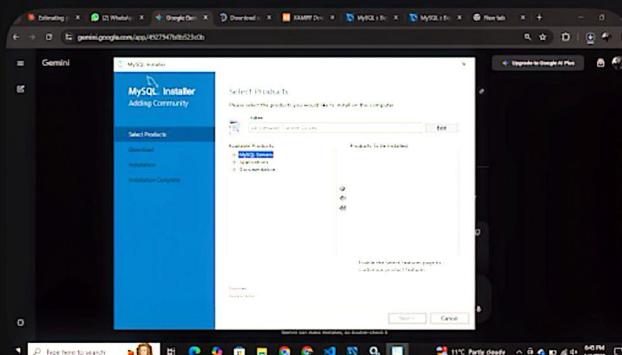


Fast





Sources



Analysis



It looks like you've successfully opened the MySQL Installer. Since the "MySQL Server" service was missing earlier, we need to install it now.

Follow these steps based on your latest screen (image\_10a2a7.png):

### 1. Select the Product to Install

1. In the "Available Products" list, click the + next to MySQL Servers.

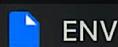


Ask Gemini



Fast



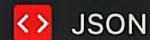


ENV

db



package



i hvae these files i want to make more of ▾  
these help me first you understand  
these then will give you the one i want...



Analysis ▾



I have thoroughly reviewed the files for your SmartSOS backend. It's a well-structured Express.js application designed to handle emergency reporting, evidence management, and administrative oversight using AWS services (S3 and SNS).

Here is a summary of the current architecture to ensure we are aligned:



### Core Architecture Summary

- **API Structure:** A modular design where server.js acts as the entry point, delegating specific features to a routes/ directory.
- **Database:** Uses mysql2 with a connection pool (db.js) to manage data for users, incidents, and evidence.



Ask Gemini



Fast

