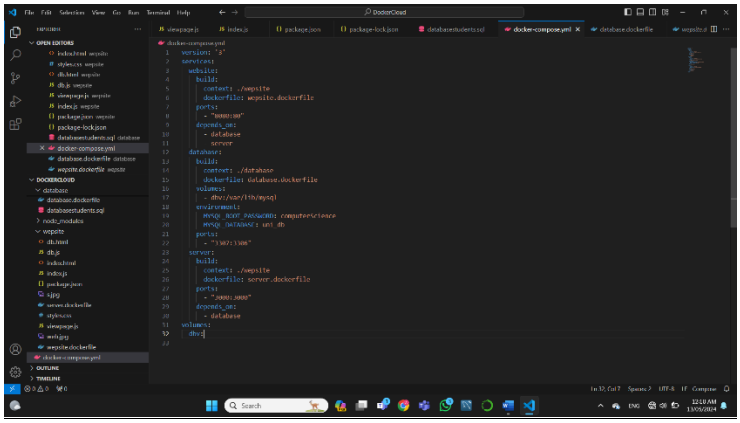
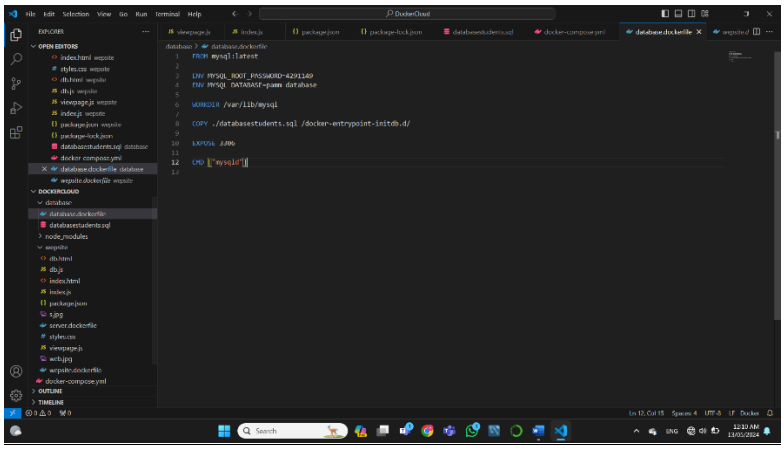
Report of cloud computing project

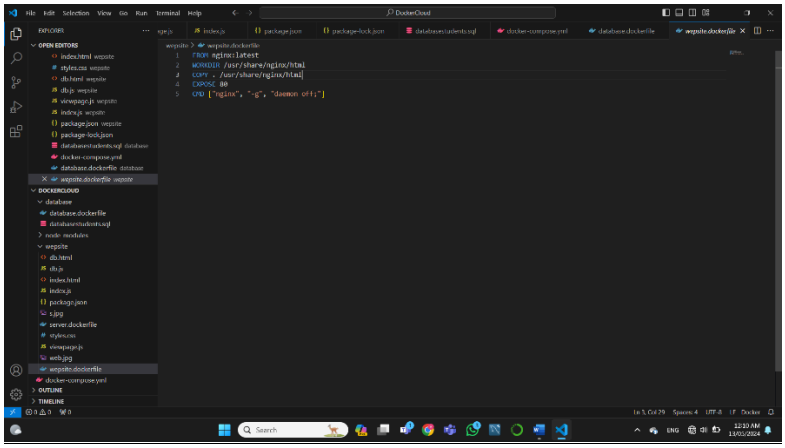
1) docker file compose



2) docker file of database



3) docker file of website



((Docker file explain))

FROM → after it the server : its version

WORKDIR→ change the working directory inside Docker container to run from this directory

COPY→ copies the content of the directory from host tp my directory in docker container and work with it

EXPOSE→ AFTER it the port number

ENV→ the envirnoments for my database

CMD→ for execute container

-g , daemon off → make my server work in the foreground

((Foreground Execution))

When a process runs in the foreground, its output (such as log messages, status updates, and errors) is displayed directly in the terminal where the process was started.

You can interact with the process by providing input and its behavior in real-time.

The process remains attached to the terminal session, and its lifecycle is tied to the terminal window. If you close the terminal, the process will also terminate.

((Background Execution (Daemon)))

when a process runs in the background (as a daemon), it detaches from the terminal and continues executing independently.

Background processes are typically started with the & symbol at the end of the command (e.g., my_process &).

They do not display output directly in the terminal, making it harder to observe their behavior.

Background processes continue running even after the terminal session is closed

Services→ defines the services (containers) that make up your app

Volumes→ This is mounts volumes to the service

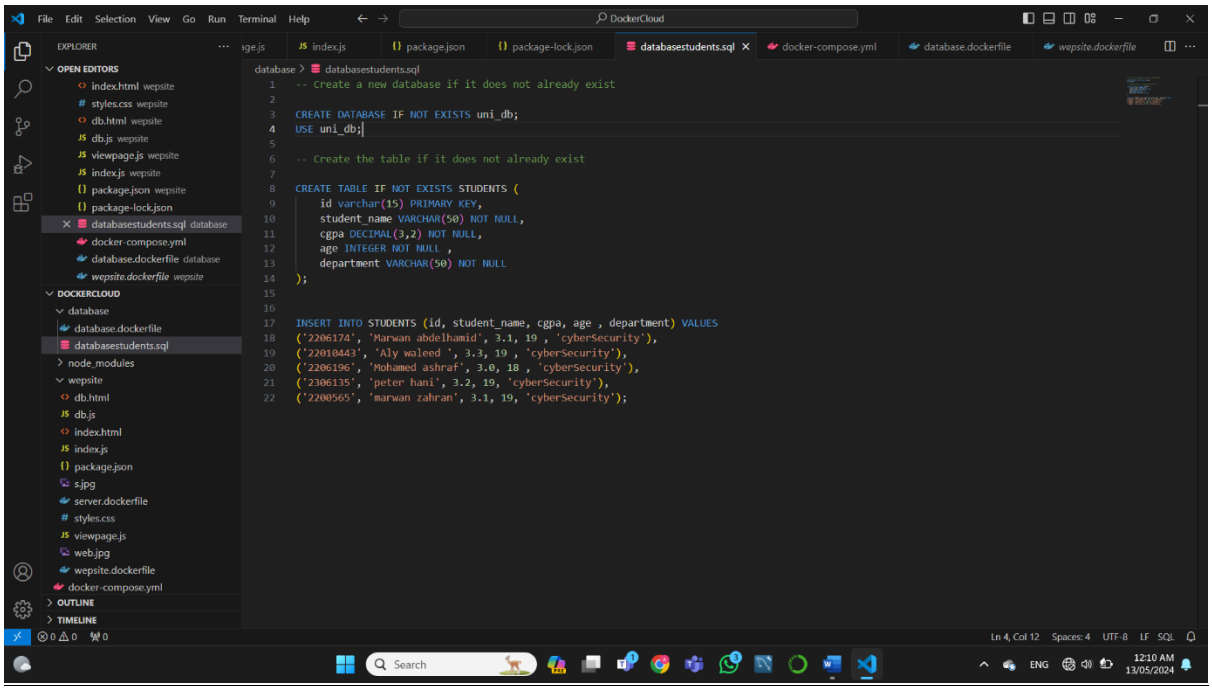
Build→this is apply the build time

Context→ sets the build context to the current directory

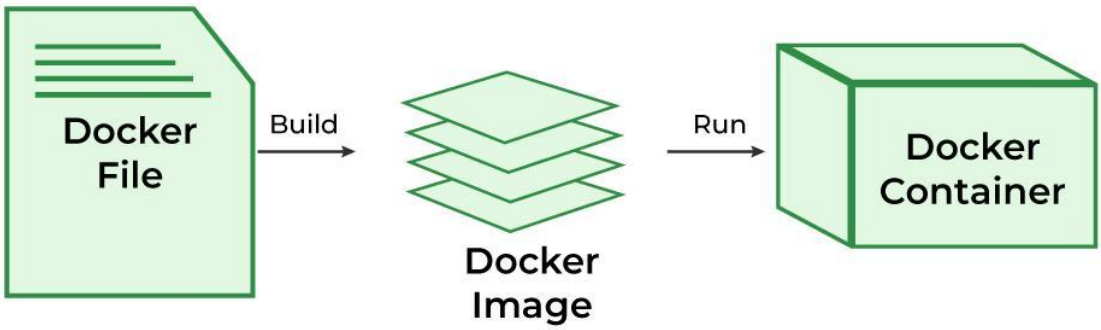
Dockerfile→specifies the name of the Dockerfile.

depends_on→to define that is the specific servies depend on and who start first

→ Database code with my sql

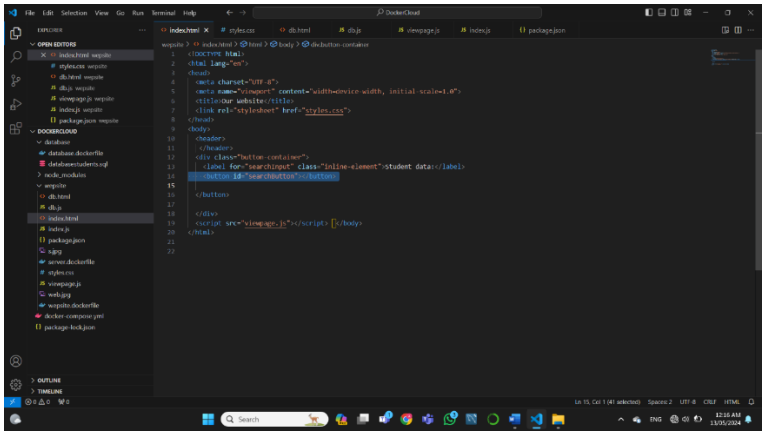


→steps to work with docker

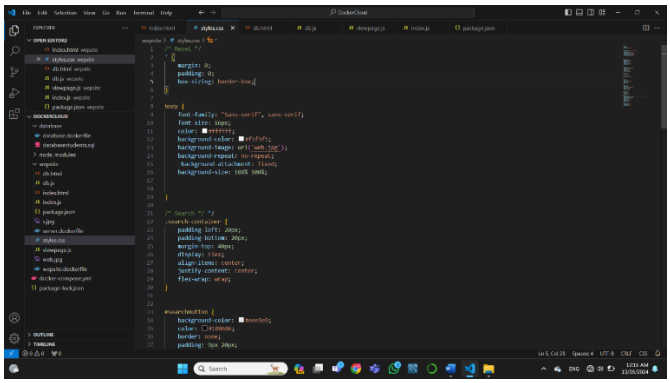


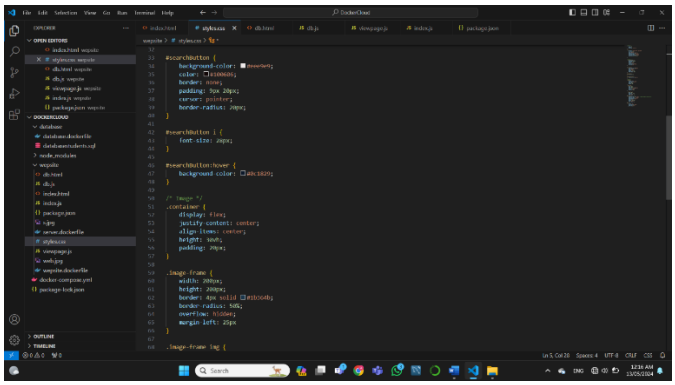
→ websites code with html ,css,javascript

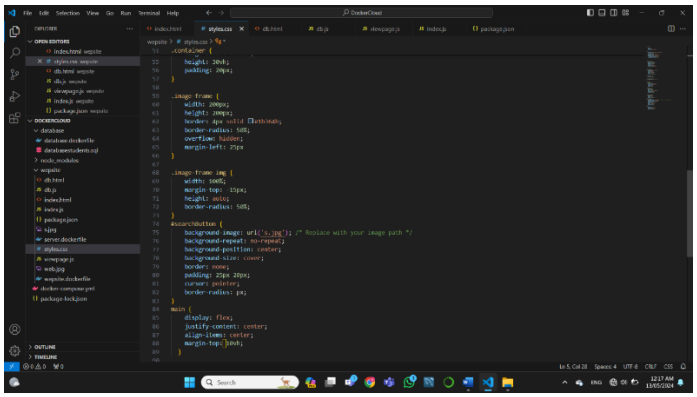
1) html for main website



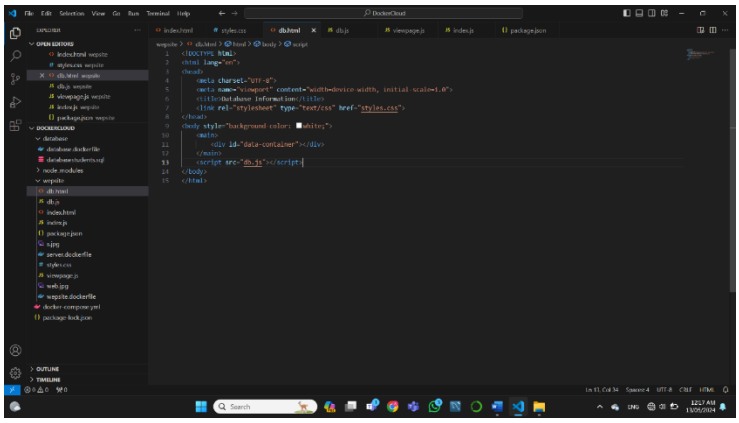
2) css is for design for my wepsite



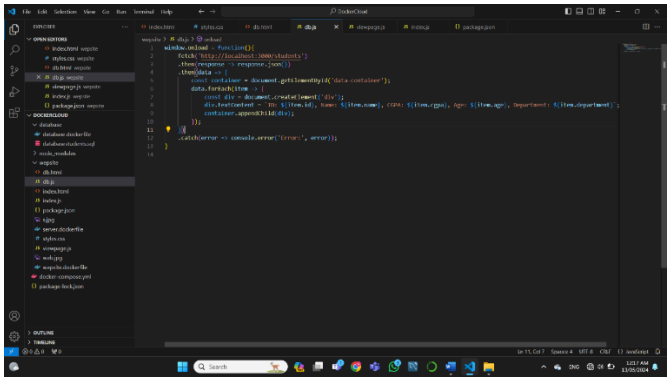




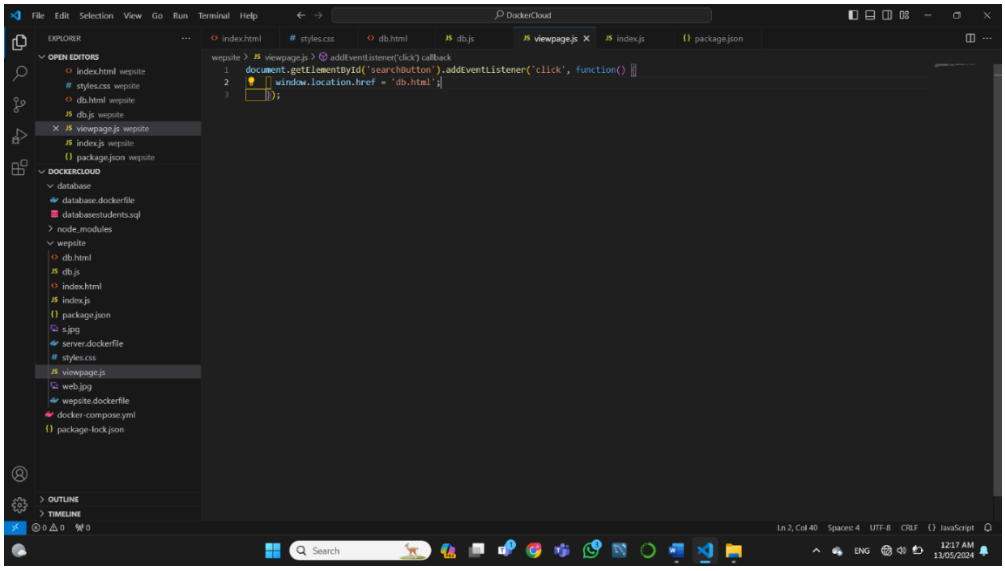
3) the html for database webpage display



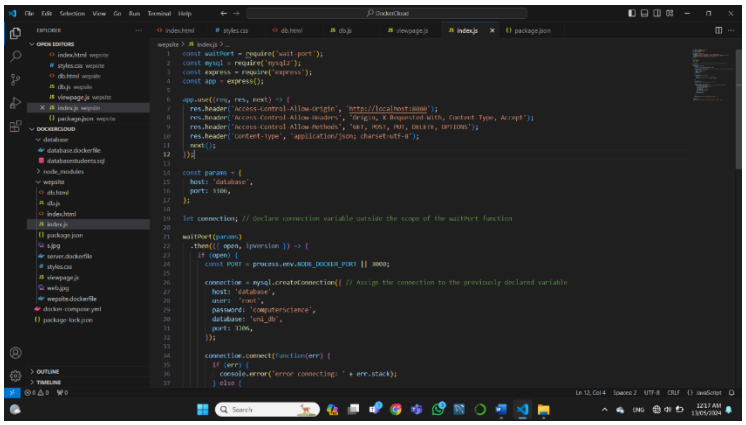
4) This JavaScript code is designed to fetch and display student data from a local server when the webpage loads

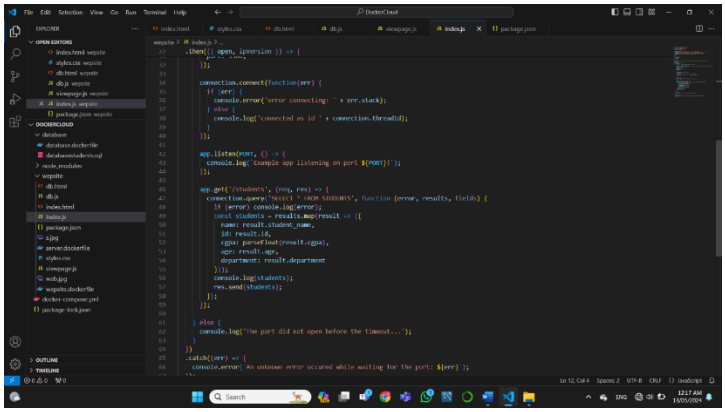


5) this JavaScript to code is when you click to the button in my main webpage open me a new page

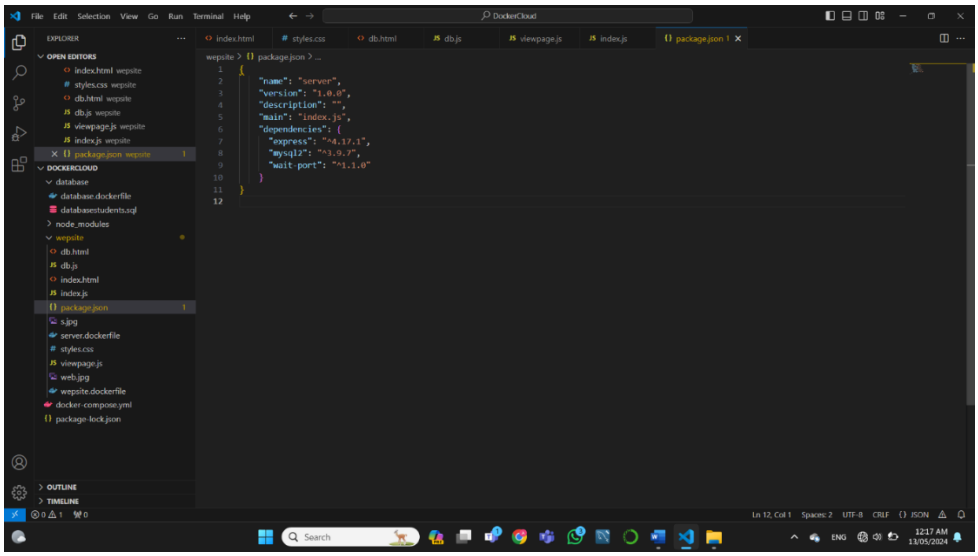


6) This JavaScript code is a simple Express.js server application that connects to a MySQL database and provides an API endpoint to fetch student data

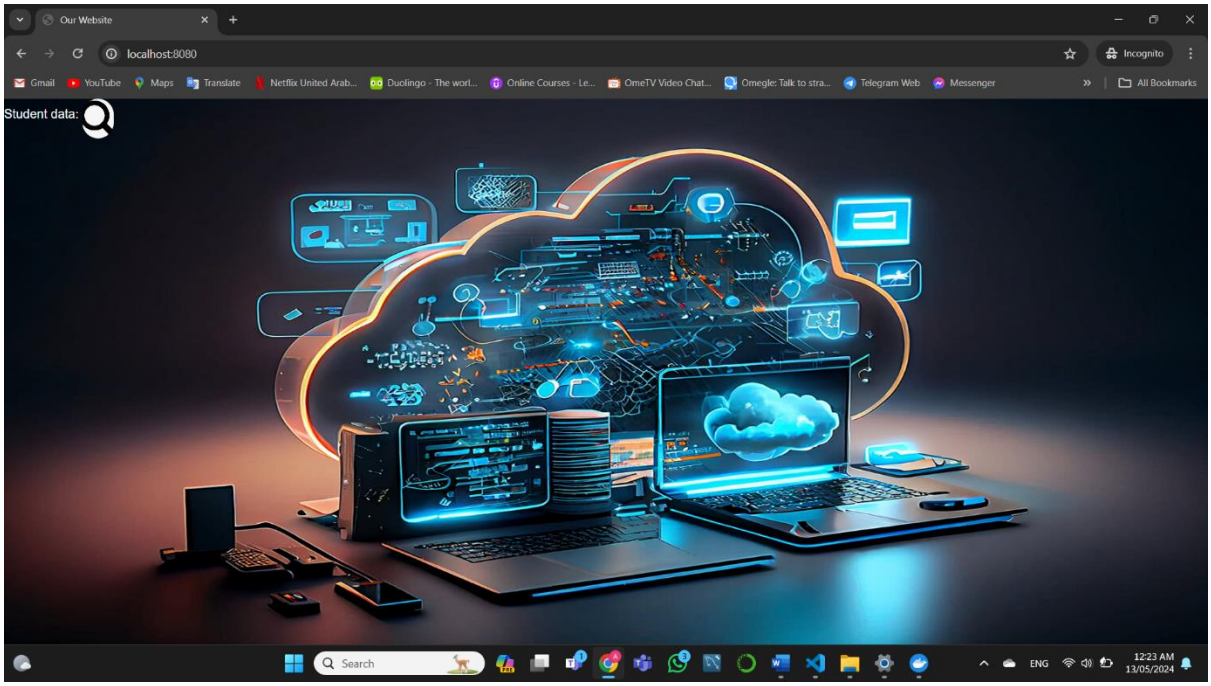




7) this is a standard file in Node.js projects. It serves as the manifest for the project and contains metadata about the project and its dependencies



→The website and compose with database



→to delete cashe

D:/DockerCloud>docker system prune -a

→to build conatiners and images for this project

D:/DockerCloud>docker-compose up --build -d