CSCI 578 – Software Architectures Fall 2025 Team Project

PocketLLM Portal

Team Number: 9

PocketLLM Portal is a lightweight web application that allows end users to interact with a compact language model running entirely on a budget CPU. The system is designed to support resource-constrained environments with careful use of caching, efficient model serving, and responsive web interactions. PocketLLM Portal meets tight resource constraints typical of entry-level cloud VMs or consumer desktops (up to 4 vCPUs and up to 16 GB RAM).

Source Code Structure

The project contains two major components: the React frontend (`client/`) and the Node.js backend (`server/`). The directory layout is as follows:

- `client/`: **React Frontend**
  - src/components/: UI Components (ChatInterface, Sidebar)
  - src/hooks/: Custom hooks (useChatStream)
  - src/context/: State management (AuthContext)
- `server/`: **Node.js Backend**
  - src/controllers/: Request handlers (ChatController)
  - src/services/: Business logic modules (InferenceService, SessionRepository)
  - src/routes/: REST API route definitions
- **Other Project Files**
  - Dockerfile: Defines the production container build
  - docker-compose.yml: Defines service startup, networking, and resource limits for running the app

Running the Project (Docker Mode)

Prerequisite:

Ensure **Docker Desktop** or **Docker Engine** is installed.

Option 1: Pulling docker image from DockerHub and running the application locally

1. Pull the Docker Image from DockerHub:

Open a terminal and run:

```
docker pull pmayurusc19/csci-578_group-9_pocket-llm:latest_final
```

This will:

- Pull the latest and final docker image for our PocketLLM Application from DockerHub
- Verify this by running the following command:
  `docker image list`

And verifying that the following image is visible in the list and note the image-id

REPOSITORY	TAG	IMAGE ID	SIZE
pmayurusc19/csci-578_group-9_pocket-llm	latest_final	42e0fc85e806	5.01GB

2. Run the PocketLLM Application

Once the docker image is pulled from DockerHub, run the following command to start the application

```
docker run -p 3001:3001 pmayurusc19/csci-578_group-9_pocket-llm:latest_final
```

This will:

- Start the container
- Launch both frontend and backend
- Download the model automatically if it is not yet present

3. Access the Web Application:

Once the container is running, open your browser and navigate to:

`http://localhost:3001`

This is the entry point to the PocketLLM portal.

4. Stop the Application:

To fully stop the container, press `Ctrl+C` in the terminal where the above Docker container is running.

To fully remove the container (and image):

Extract the container-id using the following command and locate the container running the above image

```
docker ps -a
```

Next stop the container using the container-id extracted above

```
docker stop <container-id>
```

Next, remove the container using

```
docker rm <container-id>
```

(Optional) Finally, to remove the image:

```
docker rmi pmayurusc19/csci-578_group-9_pocket-llm:latest_final
```

Option 2: Building docker image locally and running the application

1. Build and Run the Application:

Open a terminal in the project root directory (where `docker-compose.yml` is located) and run:

```
docker-compose up --build
```

This will:

- Build the docker image
- Start the container
- Launch both frontend and backend
- Download the model automatically if it is not yet present

2. Access the Web Application:

Once the container is running, open your browser and navigate to:

```
http://localhost:3001
```

This is the entry point to PocketLLM portal.

3. Stop the Application:

Press `Ctrl+C` in the terminal where Docker compose is running.

To fully stop and remove the container:

```
docker-compose down
```

Running the Project (Local Development Mode)

a) Start the Backend:

```
cd server
```

```
npm install
```

```
npm start
```

Backend runs on:

`http://localhost:3001`

b) Start the Frontend:

```
cd client
```

```
npm install
```

```
npm run dev
```

Frontend runs on:

`http://localhost:5173`

API calls will be proxied to 3001.

Enabling Real AI (Hybrid Mode)

By default, the system uses a **Mock Inference Engine** for development.

To enable full inference using a **quantized Llama** model:

1. Download the Model:

Run:

```
node download_model.js
```

This script downloads:

- **Llama-3.2-1B-Instruct-Q4_K_M (≈800MB)**

This will save the GGUF model to `server/models/`.

2. Restart the Backend:

Restart via Docker or Local mode.

The backend will automatically detect the model file and switch to **Real Inference Mode**.

Docker Deployment: Extended Instructions & Verification

The following steps help you verify whether your application is successfully running inside Docker:

1. Run in Background (Detached Mode)

```
docker-compose up -d --build
```

This builds the image, starts the container, and keeps it running in the background.

2. Verify Connectivity

`http://localhost:3001`

You should see the Pocket LLM Portal UI.

3. Ensuring the Server Runs Inside the Container

Step A — List Running Containers

```
docker-compose ps
```

Copy the container name (e.g., `saimplementation-pocket-llm-1`).

Step B — Enter the Container

```
docker exec -it saimplementation-pocket-llm-1 /bin/bash
```

If `/bin/bash` is unavailable, try `/bin/sh` instead.

Step C — Check Running Node Processes

Inside the container, run:

```
ps aux | grep node
```

You should see:

- The `npm start` process

- The actual server process (`node src/app.js` or similar).

Step D — View Live Logs (Alternative to Entering the Container)

```
docker-compose logs -f
```

Press `Ctrl + C` to exit logs.