# ChatbotAI – Project Summary

This project is a simple AI chatbot web application. It allows users to log in, send questions, and receive AI responses. All conversations are saved and shown back to users when they return.

## What it does

* User can sign up and log in securely.
* User sends a message and gets a reply from an OpenAI (mock).
* Conversations are saved in a SQL Server database.
* Displays all past messages when the user logs in again.

## How it works

1. User logs in.
2. User sends a question.
3. The frontend sends it to the backend.
4. The backend contacts OpenAI (mock) or returns a sample reply.
5. The reply and the user message are saved in the database.
6. Both are displayed in the chat interface.

## Tech Stack

**Backend** – ASP.NET Core 8

* C# Web API
* Entity Framework Core
* SQL Server
* Follows SOLID principles
* Repository pattern used

**Frontend** – React + TypeScript

* CSS for layout and basic styling
* Axios for HTTP calls
* React Router for navigation

## Database Structure

The app stores users and chat messages in a SQL Server database. Here's the structure:

-- Create the database

CREATE DATABASE ChatbotAI;

GO

USE ChatbotAI;

GO

-- Users table

CREATE TABLE Users (

Id INT PRIMARY KEY IDENTITY(1,1),

Username NVARCHAR(100) NOT NULL UNIQUE,

Email NVARCHAR(255),

PasswordHash NVARCHAR(255) NOT NULL,

Role NVARCHAR(50) DEFAULT 'User',

IsActive BIT DEFAULT 1,

CreatedAt DATETIME DEFAULT GETDATE()

);

GO

-- Messages table

CREATE TABLE Messages (

Id INT PRIMARY KEY IDENTITY(1,1),

UserId INT NOT NULL,

MessageText NVARCHAR(MAX) NOT NULL,

ResponseText NVARCHAR(MAX),

CreatedAt DATETIME DEFAULT GETDATE(),

FOREIGN KEY (UserId) REFERENCES Users(Id)

);

GO

## How to Run

### Backend (ASP.NET Core)

cd ChatbotAI\Backend

dotnet restore

dotnet ef database update

dotnet run

### Frontend (React)

cd ChatbotAI\Frontend

npm install

npm start

The frontend will start on http://localhost:3000 and connect to the backend API.

## Folder Structure

ChatbotAI

│

├── Backend

│ ├── Controllers

│ ├── DTOs

│ ├── Models

│ ├── Repositories

│ ├── Services

│ └── Data (DbContext)

│

└── Frontend

├── api

├── components

├── pages

├── services

├── types

└── App.tsx

## Notes

* Passwords are stored as is (mock).
* Login is basic and meant for learning or demo purposes.
* Chat UI is a simple list of user messages and AI responses.
* No external styling libraries (just CSS).
* AI responses are mocked.