AI Chatbot Web App — Development Checklist (University Project)

1. Project Setup

- Define chatbot purpose (e.g. student Q&A;, study helper).
- Choose tech stack: Frontend (HTML/CSS/JS or React), Backend (Flask or FastAPI), Database (SQLite), AI Model (LLaMA via Ollama).
- Create project folder & virtual environment.
- Initialize Git repo (optional).
- Write a short README.md explaining purpose & tech used.

2. Environment Setup

- Install Python (3.10+).
- Install dependencies: pip install flask requests.
- Install Ollama from https://ollama.ai/download.
- Pull your model: ollama pull llama3.
- Test model: ollama run llama3.

3. Backend Development

- Create app.py and implement /chat route.
- Accept JSON { 'message': '...' } and call local Ollama API (http://localhost:11434/api/generate).
- Return chatbot’s reply in JSON.
- Optional: /history or /login routes.
- Enable CORS if needed for frontend requests.

4. Database (Optional)

- Use SQLite (no setup needed).
- Create chat_history.db with tables: users and messages.
- Add helper functions to save and retrieve messages.
- Verify persistence between sessions.

5. Frontend Development

- Create index.html with chat window, input, and send button.
- Use fetch() to call /chat endpoint.
- Dynamically display messages in the DOM.
- Add minimal CSS styling.
- Test full chat flow end-to-end.

6. Integration Testing

- Run Ollama: ollama run llama3.
- Start Flask backend: python app.py.
- Open index.html and test chat interaction.
- Ensure backend, model, and frontend communicate correctly.
- Add error handling for missing model/server issues.

7. Security & Data Handling

- Keep Ollama on localhost only.
- Sanitize inputs before sending to backend.
- Hash passwords if login is included.
- Avoid storing sensitive data.

8. Documentation & Submission

- Write README.md with setup, description, and screenshots.
- Add code comments for clarity.
- Organize files: app.py, index.html, database, static/, README.md.
- Zip folder for submission or upload to GitHub.

9. Optional Bonus Features

- Save and display chat history.
- Add login system.
- Allow model selection (llama3, mistral, etc.).
- Add dark/light mode or clear chat option.
- Show timestamps and system prompts.

Final Submission Checklist

- Frontend works in browser.
- Backend responds correctly.
- Ollama model runs locally.
- Database stores chat (if used).
- README includes setup instructions.
- Project runs cleanly on another machine.