Database & Deployment Details

ER Diagram

```
erDiagram
    ASSESSMENTS {
        UUID id PK
        DATETIME created_at
        TEXT attack_type
        TEXT status
        UUID original_pdf_id FK
        UUID answers_pdf_id FK
        UUID attacked_pdf_id FK
        UUID report_pdf_id FK
    }
    STORED_FILES {
        UUID id PK
        TEXT path
        TEXT mime_type
        DATETIME uploaded_at
    }
    QUESTIONS {
        INTEGER id PK
        UUID assessment_id FK
        INTEGER q_number
        TEXT stem_text
        JSONB options_json
        TEXT gold_answer
        TEXT gold_reason
        TEXT attacked_stem
    }
    LLM_RESPONSES {
        INTEGER id PK
        INTEGER question_id FK
        TEXT model_name
        TEXT llm_answer
        TEXT llm_reason
        JSONB raw_json
        DATETIME created_at
    }

    ASSESSMENTS ||--o{ QUESTIONS : has
    ASSESSMENTS }o--|| STORED_FILES : original_pdf_id
    ASSESSMENTS }o--|| STORED_FILES : answers_pdf_id
    ASSESSMENTS }o--|| STORED_FILES : attacked_pdf_id
    ASSESSMENTS }o--|| STORED_FILES : report_pdf_id
```

```
QUESTIONS ||--o{ LLM_RESPONSES : has
```

Database Initialization & Migration

- **Initialize the database:**

  ```
  cd backend
  flask db upgrade
  ```

- **Create a new migration after changing models:**

  ```
  flask db migrate -m "Describe your change"
  flask db upgrade
  ```

- **Configuration:**
  - The database URI is set via the `DATABASE_URL` environment variable (see `.env`).
  - Default: `postgresql+psycopg://localhost/ftai`

Accessing the Database

- **Directly via psql:**

  ```
  psql <your-db-url>
  ```

- **With a GUI:**
  - Use DBeaver, TablePlus, or pgAdmin with the same `DATABASE_URL`.

- **With SQLAlchemy in a script:**

  ```
  from app import db, create_app
  app = create_app()
  with app.app_context():
      # Now you can use db.session, db.engine, etc.
  ```

Using the DB from Docker

- **Run Postgres in Docker:**

  ```
  docker run --name fairtestai-db -e POSTGRES_PASSWORD=yourpassword -e POSTGRES_DB=ftai -
  ```

- **Set your `DATABASE_URL` in `.env` or environment:**

  ```
  DATABASE_URL=postgresql+psycopg://postgres:yourpassword@localhost:5432/ftai
  ```

- **Run migrations:**

```
cd backend
flask db upgrade
```

- **Now your Flask app can connect to the Dockerized DB.**

How the DB is Used in the App

- **On Upload:**
  - Uploaded PDFs are saved as `StoredFile` entries.
  - An `Assessment` is created, linking to the uploaded files.
  - Questions are parsed and stored as `Question` entries.
- **On Processing:**
  - Attacked and report PDFs are generated and saved as new `StoredFile` entries.
  - The `Assessment` is updated to link to these new files.
  - LLM responses (if any) are stored as `LLMResponse` entries.

Summary Table

Task	Command/Action
Initialize DB	`flask db upgrade`
Create migration	`flask db migrate -m "desc"`
Apply migration	`flask db upgrade`
Access DB (psql)	`psql <db-url>`
Access DB (GUI)	Use DBeaver/TablePlus/pgAdmin with `DATABASE_URL`
Use DB from Docker	Run Postgres container, set `DATABASE_URL`, run migrations
Use DB in script	Use `create_app()` and `app.app_context()`

Example Docker Compose (Optional)

```yaml
version: '3.8'
services:
  db:
    image: postgres:15
    environment:
      POSTGRES_PASSWORD: yourpassword
      POSTGRES_DB: ftai
    ports:
      - "5432:5432"
```

```
    volumes:
      - db_data:/var/lib/postgresql/data
  backend:
    build: ./backend
    environment:
      DATABASE_URL: postgresql+psycopg://postgres:yourpassword@db:5432/ftai
      OPENAI_API_KEY: your-openai-key
    depends_on:
      - db
    ports:
      - "5000:5000"
volumes:
  db_data:
```