EduAssist Backend – Comprehensive ReadMe

🚀 Project Overview

EduAssist is an AI-powered teaching assistant platform that helps educators and students with automating Q&A, creating study guides, generating assessments, building slide decks, managing requests, and collecting/analysing feedback. The backend is a layered modular monolith built with FastAPI, async SQLAlchemy, PostgreSQL, and GenAI integrations. Authentication is secure with JWT tokens, and email/service features use free providers (like SendGrid).

📚 User Stories → Features Mapping

Feature	Primary User Stories
Knowledge Assistant	US 3, 4, 6: Instant answers to student Qs using course documents; Confidence scoring and escalation; Conversation history
Study Guide Generator	US 1, 2: Generate study guides from YouTube/video or notes; User sets segments/priorities; Feedback and confusion heatmap
Assessment Generator	US 2, 6, 10: TA creates assessments from material; MCQ/MSQ/NAT selection and auto-grading; Student feedback on questions
Slide Deck Generator	US 1, 2: Build slides from notes/docs; TA customizes style, depth, and export; Embedded student feedback at slide/foot/deck level
Admin Workflow Agent	US 5, 8, 9: Process extension/objection requests; AI agent checks history, recommends outcome, drafts responsive emails
Feedback System	US 5+ for each feature: Collect ratings/comments per assessment, study guide, slide deck & AI responses; TA dashboard for feedback review

🏗️ Backend Components

API Layer

- **Routers per feature**: `/knowledge-assistant`, `/study-guides`, `/assessments`, `/slide-decks`, `/admin-workflow`, `/feedback`, `/auth`
- CRUD endpoints for create/read/update/delete objects and requests

Service Layer

- **KnowledgeAssistantService**: orchestrates RAG, LLM, vector search, chat
- **StudyGuideService**: extracts CC/captions, segments video, generates prioritized guide
- **AssessmentGeneratorService**: Gemini generates, difficulty calibrator, grading
- **SlideDeckService**: chunks text, outlines session, builds Markdown slides
- **WorkflowAgentService**: multi-step agent, reviews requests, sends draft emails
- **FeedbackService**: stores feedback, triggers TA notifications
- **MailService**: password resets, notifications (free API)

Data Layer

- **Async SQLAlchemy Models:**
  - `User`, `Course`, `Document`, `Conversation`, `Message`, `Assessment`, `Question` (bank), `AssessmentAttempt`, `StudyGuide`, `VideoSegment`, `SlideDeck`, `Slide`, `WorkflowRequest`, `Feedback`, etc.
- **ChromaDB for RAG/Vector Search**: Fast context retrieval

Auth & Security

- JWT tokens with expiry; bcrypt password hashes
- Role-based permissions (Student, TA, Admin)
- Password reset with email link (SendGrid free tier)
- CORS, rate limits, logging, error handlers

GenAI Integration

- Gemini API for responses, generation, embedding

- LangChain for multi-step, tool-using workflows (admin only)
- ChromaDB for semantic/retrieval augmentation

📈 Feature ↔ User Story Connection Table

Feature	Example User Story	API Route(s)
Knowledge Assistant	"As a student, I want instant, accurate answers on course docs"	`/knowledge-assistant`
Study Guide Generator	"As a student, I want to highlight segments and get focused guides"	`/study-guides`
Assessment Generator	"As a TA, I want to auto-create assessments and grade results"	`/assessments`
Slide Deck Generator	"As a TA, I want slides for each topic, with TA and student feedback"	`/slide-decks`
Workflow Agent	"As a TA/Admin, I need requests auto-analyzed and response drafted"	`/admin-workflow`
Feedback System	"As a student, I want to rate and comment on each AI feature"	`/feedback`

🧪 Basic Tests Currently Added

- **Auth:** registration, login, password reset, JWT checks
- **Knowledge Q&A:** conversation creation, doc upload, chat round-trip, response correctness
- **Study Guide:** guide creation from YouTube, segment priority, guide fetch, feedback submit
- **Assessment:** assessment creation, grading, attempt/submit workflow, question feedback
- **Slide Deck:** deck creation, slide outline, edit/update, feedback recording
- **Workflow Agent:** extension request simulation, agent recommendation, email draft
- **Feedback:** submit rating/comment for guide/assessment/etc.; list per object

🛠️ Installation & Setup

1. Clone & Setup Python Env

```
git clone <repo>
cd backend
python3.12 -m venv .venv
source .venv/bin/activate
```

2. Install dependencies

```
pip install -r requirements.txt
```

3. .env configuration (example)

```
SECRET_KEY=your-secret-key-here
ALGORITHM=HS256
SENDGRID_API_KEY=your_free_sendgrid_api_key
EMAIL_SENDER=your_verified@email.com
FRONTEND_URL=https://your-frontend.com
DATABASE_URL=postgresql://user:pass@<neon>.neon.tech/db?ssl=require
GEMINI_API_KEY=...
CHROMA_PERSIST_DIRECTORY=./chroma_db
```

4. Run Database Setup (no Alembic, auto-create)

See `app/database.py` for auto-create call on startup (or run `Base.metadata.create_all()` before first use).

5. Start Server

```
uvicorn app.main:app --reload
```

6. Run Tests

```
pytest tests/ -v
```

🎯 Features at a Glance

- **Q&A Knowledge Assistant:** RAG chatbot, document upload, context-aware, confidence scoring, escalation
- **Study Guide Generator:** YouTube/video section tagging, CC caption extraction, prioritized summary, student feedback
- **Assessment Generator:** AI-generated questions from notes/docs, MCQ/MSQ/NAT, auto-grading, student feedback
- **Slide Deck Creator:** AI-built slides from notes/docs, editing/outlining, markdown export, student feedback
- **Admin Workflow Agent:** Extension/objection auto-analysis, LangChain agentic workflow, draft responses
- **Feedback System:** Ratings/comments on every main feature, TA dashboard, analytics
- **Security:** JWT, RBAC, password/email workflows
- **Deployability:** All supports Vercel/Render deploys, Neon DB, free-tier use

📦 Folder Structure

```
app/
  - main.py
  - routers/ (per feature)
  - services/ (feature business logic)
  - models.py
  - schemas/
  - database.py
  - crud.py
  - auth/
  - config.py
  - tests/
```

💡 Notes

- For production: set up email senders on SendGrid/Mailgun and validate domain
- To update database schema, re-run auto-create or use Alembic migrations
- Use OpenAPI/Swagger UI at `http://localhost:8000/docs` for API exploration

- Feedback and analytics features rely on proper frontend integration for value

---

Maintained by Team 38 – SE Sep 2025