TutorTech Platform Configuration & Deployment Guide

This handbook is intended for developers or technical maintainers assuming ownership of the TutorTech AI-powered learning platform. It explains how to configure, deploy, and maintain every core aspect of the system: frontend, backend, vector and relational databases, and hosting environments. This will also help you make secure and scalable edits, including updating environment variables, customizing AI behavior, or managing user and course data.

Project Structure Overview

TutorTech/

├── frontend/ # React.js application

├── backend/ # Flask-based API server

├── .env # Environment variables (locally)

├── public/assets # Static images and styles

├── database/ # SQL scripts for PostgreSQL schema

Backend Configuration (Flask, OpenAI, Qdrant, PostgreSQL)

1. Environment Variables (.env)

The backend uses a .env file to manage secrets:

OPENAI_API_KEY=sk-...

QDRANT_HOST=https://your-qdrant-instance

QDRANT_API_KEY=your-qdrant-key

DB_HOST=localhost

DB_NAME=tutortech_db

DB_USER=youruser

DB_PASSWORD=yourpassword

DB_PORT=5432

Changing the environmental variables on Render:

- Go to **Render Dashboard > Environment > Secret Files or Variables**.
- Update variables (like OPENAI_API_KEY) with your new variables.
- Re-deploy or restart the service.

2. Flask App Entry Point

The backend is in app.py:

- Connects to PostgreSQL (psycopg2) and Qdrant
- Embeds prompts using OpenAI Embeddings (text-embedding-ada-002)
- Routes: /login, /signup, /chat, /courses, /assignments, /grades, etc.
- CORS is enabled for cross-origin access from the frontend

3. Qdrant Vector DB

- Initializes a chat_history collection for semantic memory.
- Chat messages are embedded and stored with timestamps.
- Deletes messages older than 30 days via timestamp filter.

To change the embedding model or distance type: Edit this line in app.py:

vectors_config=VectorParams(size=1536, distance=Distance.COSINE)

Frontend Configuration (React.js)

1. Environment Variables

Create a .env file at the root of frontend/:

REACT_APP_API_URL=https://your-backend-url.onrender.com

This allows all API calls in files like Assignment.js, Chat.js, and UserContext.js to function correctly.

To change the backend connection:

- Update REACT_APP_API_URL in .env
- Run npm run build and redeploy

2. Important Components

- UserContext.js - maintains login state and chat history toggle
- Dashboard.js - toggles chat history and links to all user tools
- Chat.js - sends prompts to Flask and receives responses
- Assignment.js - AI-graded assignment submissions
- LearningStyleQuiz.js - stores preferences to personalize AI behavior

3. Hosting on Vercel

1. Link your GitHub repo in Vercel
2. Set **Environment Variables** in Project Settings:
    - REACT_APP_API_URL=https://your-backend-url.onrender.com
3. Use default npm run build command
4. Ensure your vite.config.js or package.json build targets React 18+

To switch GitHub repo for Vercel hosting:

- Go to **Vercel Dashboard > Project Settings > Git**
- Click "Disconnect Git Repository"
- Connect a new GitHub repo and redeploy

PostgreSQL Database (Relational)

1. Initial Schema

Defined in TutorTech_db_demo.sql, includes:

- student_information
- course_information
- course_modules, course_lectures, course_assignments
- grades, assignment_results, enrollments

2. Connecting Remotely

Make sure your Render PostgreSQL service allows access from your backend:

- Use the **Internal Database URL** format in app.py

- Alternatively, expose public IPs and use SSL connections for local testing

To change DB credentials: Update .env on Render:

DB_HOST=...

DB_USER=...

DB_PASSWORD=...

User Authentication and Session

Sessions

- Flask sets a secure HTTP-only cookie user_id
- Frontend uses this cookie to persist login session across refreshes (see UserContext.js)
- Protected routes in React (ProtectedRoute.js) ensure only logged-in users access internal pages

---

AI Chat + Learning Styles

Modes

- Tutor, Mentor, Co-Learner = system presets (via bot_prompts)
- Custom = loaded from DB via /get-preferences

Stored in student_information.learning_preferences


```
{
  "response_length": "short",
  "guidance_style": "real_world",
  "value_focus": "direct"
}
```


To edit AI behavior:

- Modify generate_prompt_from_preferences() in app.py
- Add or modify preset bot prompts in bot_prompts

Advanced AI Behavior Customization

- **To change how semantic search works**, edit the logic in chat() in app.py, including:
    - similarity thresholds (e.g., score_threshold=0.75)
    - fallback behavior if no relevant messages are found
- **To change conversation pairing behavior for vague prompts**, update the is_vague_prompt() function and pairing injection logic inside chat()
- **To adjust how messages are prioritized**, edit the merge_histories() function

---

Deployment Steps (Render + Vercel)

Backend (Render)

1. Create a **Web Service**
2. Use Python 3.10+, Gunicorn for production
3. Add environment variables
4. Use build command: pip install -r requirements.txt
5. Use start command: gunicorn app:app

Frontend (Vercel)

1. Connect GitHub repo
2. Set REACT_APP_API_URL
3. Set framework to React
4. Deploy

Testing & Debugging

Testing Chat Locally

- Use Postman or curl to POST to localhost:5000/api/chat
- Check embeddings and payload in Qdrant dashboard

Debugging Tips

- Watch Render logs for Flask errors
- Watch Vercel's logs for frontend build issues
- Check Qdrant console for malformed embeddings or timeouts

Updating the Platform

Updating AI Behavior

- Modify prompt templates or system messages in app.py
- Add new modes by expanding bot_prompts

Changing Assignment Questions

- Insert new questions via SQL into assignment_questions
- Options/answers are structured as:

```
{
  "choices": ["A", "B", "C", "D"]
}
```

Modifying Course Content

- Use SQL to modify:
    - course_information
    - course_modules
    - course_lectures

Final Notes

For maintainers inheriting the project:

- Ensure secrets are stored securely on Render/Vercel
- Review and understand app.py logic
- Maintain consistency in environment variable naming across frontend/backend
- Confirm your database schema matches the latest frontend expectations

This document should allow new developers to fully understand, deploy, and evolve the TutorTech platform with minimal guesswork.