

App Summarizer

A full-stack, production-ready summarization web app with:

- **Backend:** Flask, SQLAlchemy, PostgreSQL, JWT Authentication, Celery, Redis (async tasks)
 - **Frontend:** React (Vite), Axios, Tailwind CSS
 - **Infrastructure:** Docker, Docker Compose
-

Features

- **Secure authentication** via JWT and/or Google OAuth
 - **AI-powered text & URL summarizer** (async background processing)
 - **User-based summary storage and history**
 - **Production-ready:** Dockerized, with multi-stage build support
 - **Modern UI:** Responsive React frontend, Tailwind CSS
-

Project Structure

```
app-summarizer-final/ | └── backend/ # Flask app, Celery worker, requirements.txt, Dockerfile | └──  
app.py | └── models.py | └── tasks.py | └── ... | └── frontend-vite/ # React (Vite), Tailwind, Axios,  
Dockerfile, public, src | └── src/ | └── package.json | └── ... | └── docker-compose.yml # Unified  
orchestration for dev/prod └── Dockerfile(s) # For backend & frontend containers
```

Tech Stack

Tech Stack

Area	Technology
Backend	Flask, SQLAlchemy, PostgreSQL, Celery, Redis, JWT
Frontend	React (Vite), Axios, Tailwind CSS
Auth	JWT (~Flask-JWT-Extended), Google OAuth
Async	Celery, Redis
DevOps	Docker, Docker Compose

Getting Started

Prerequisites

- Python 3.9+
- Node.js v18+

- PostgreSQL (locally, or Render/cloud)
- Redis (locally, or with Docker/Render)

Backend Setup

Getting Started

Prerequisites

- Python 3.9+
- Node.js v18+
- PostgreSQL (locally, or Render/cloud)
- Redis (locally, or with Docker/Render)

Backend Setup

```
cd backend python3 -m venv venv source venv/bin/activate pip install -r requirements.txt cp .env.example .env # Fill out DB and secret keys python app.py # Start Flask backend celery -A tasks worker --loglevel=info # (in new terminal tab)
```

Frontend Setup

```
cd frontend-vite npm install npm run dev # Runs local dev server on :5173
```

Docker Compose Quickstart

Alternatively, use Docker Compose for full-stack orchestration:

```
docker-compose up --build
```

Usage

1. Visit <http://localhost:5173>
 2. Sign up or log in (via JWT or Google OAuth)
 3. Paste any text or a URL, and click "Summarize"
 4. The backend handles processing via Celery (async)
 5. View, edit, or delete summaries from your account history
-

Environment Variables

- `DATABASE_URL` - PostgreSQL connection string
- `JWT_SECRET_KEY` - JWT signing key
- `CELERY_BROKER_URL` - Redis URL (e.g. `redis://localhost:6380/0`)
- `CELERY_RESULT_BACKEND` - Redis URL
- `GOOGLE_OAUTH_CLIENT_ID`, `GOOGLE_OAUTH_CLIENT_SECRET` (optional)

Set these either in `.env` files or as environment variables.

Main API Endpoints

- **POST /api/signup:** Register new users
 - **POST /api/login:** Login, returns JWT
 - **POST /api/summarize:** Submit text for async summarization (JWT required)
 - **GET /api/summaries:** Fetch user summaries
-

Project Timeline (Example)

Day	Task	Deliverable
1-7	Flask API, JWT, PostgreSQL integration	Auth, CRUD, Flask endpoints
8-14	React (Vite), login/signup, Tailwind	UI, API integration, JWT storage
15-18	Celery, Redis, dockerization	Async, productionized stack
19-21	Integration, Nginx, deployment, docs	Full cloud deployment, README

Troubleshooting

- **Too many redirects?** — Clear cookies, check OAuth, or JWT auth setup.
 - **Google OAuth invalid_scope?** — Use <https://> scopes, not <http://>.
 - **Celery/Redis issues?** — Ensure both are running and using the same port.
 - **Database errors?** — Double-check `DATABASE_URL` and DB permissions.
-

Appendix: Extended Features / Future Work

- NLP model integration
 - User organization roles
 - Extended summary formats
 - (... and more, as your roadmap evolves!)
-

Ready to submit!

Let me know if you need further custom sections or want help with a PDF/Word export, or attaching diagrams/screenshots.