💪 BodyTrust AI

Earn trust in health like you do in finance – with data.

📘 Project Overview

BodyTrust AI is an intelligent health-scoring platform that leverages machine learning, physical activity tracking, and role-based access control to provide users and admins with:

- 📊 Personalized health insights
- 🔐 Secure login and registration
- 👑 Admin dashboards and control tools
- 🔁 Real-time BodyTrust Score predictions

Built for hackathons, fitness startups, and data-driven health solutions.

⚙️ System Architecture

- **Frontend**: Dash (Plotly)
- **Backend**: FastAPI
- **Database**: PostgreSQL
- **ML Model**: RandomForestRegressor (trained on Fitbit dataset)
- **Authentication**: JWT + bcrypt hashing
- **Deployment-ready**: With `.env.example` and modular code

🤖 Algorithm & Data Pipeline

Step 1: Data Processing

- Data cleaned & merged from multiple CSVs
- Features selected: Calories, Sleep, BMI, Steps, Minutes active

Step 2: BodyTrust Score Logic

- Feature normalization
- Weighted score generation + clustering (KMeans)
- Health Tier Classification: `Elite Performer`, `Average Active`, `Needs Improvement`

Step 3: ML Modeling

- Trained RandomForestRegressor to predict future scores
- Model saved as `bodytrust_model.pkl`

Step 4: Smart Dash + Admin Logic

- Role detection via JWT
- Admin-only visibility
- Promote-to-admin & user table integration

🔚 File Structure & Notebook Map

File / Notebook	Description	Connected To
`notebook_1_cleaning.ipynb`	Cleans and merges raw Fitbit data	Dash, ML model training
`train_model.ipynb`	Trains RandomForest, saves model `.pkl`	`api_server.py` (predict route)
`dash_app.py`	Dash frontend app with user/admin views	FastAPI, ML output
`api_server.py`	FastAPI backend with JWT & prediction API	Dash frontend
`auth.py`	Password hashing & JWT token utilities	FastAPI login/register
`database.py`	SQLAlchemy model for User + DB session	FastAPI, admin logic
`dependencies.py`	Role-based guards (`admin_required` , etc.)	FastAPI
`.env.example`	Example env file (safe for GitHub)	Local setup

🔐 .env Setup & Security

Your real `.env` should never be committed. Use `.env.example` like this:

```
SECRET_KEY=your_secret_key
DATABASE_URL=postgresql://username:password@localhost/bodytrust_users
EMAIL_HOST=smtp.example.com
EMAIL_USER=your_email@example.com
EMAIL_PASS=your_password
```

🚀 Setup Instructions

```
# 1. Clone repo
https://github.com/YOUR_USERNAME/bodytrust-ai.git

# 2. Create virtual environment & install dependencies
pip install -r requirements.txt

# 3. Setup PostgreSQL DB & update .env

# 4. Start FastAPI backend
uvicorn api_server:app --reload

# 5. Run Dash app
python dash_app.py
```

👑 Admin Features

- View all registered users
- Promote user to admin
- Role-based panel visibility
- Score monitoring (live from DB)

📬 Future Features (Optional)

- Email alerts for low BodyTrust Scores
- Password reset + email verification
- Score history charts (TimeSeries)
- Health Recommendations via LLM

🙌 Built With Love For

- PhysTech 2025 Hackathon
- Fitness and health tech innovation
- Data lovers & full-stack dreamers 💙