Backend Development – Useful Commands Cheat Sheet

This file is a **single source of truth** for running, developing, testing, and maintaining your FastAPI backend. You should rarely need to ask again once this is in place.

🚀 Run the Application

Development (auto-reload – nodemon equivalent)

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

Development (limit reload to app directory – faster)

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

Production (no reload)

```
uvicorn app.main:app --host 0.0.0.0 --port 8000 --workers 4
```

🛕 Testing

Run all tests

```
pytest
```

Run tests with verbose output

```
pytest -v
```

Run a single test file

```
pytest tests/test_auth.py
```

Run tests with coverage

```
pytest --cov=app --cov-report=term-missing
```

🗼 Database & Migrations (Alembic)

Check current revision

```
alembic current
```

Create a new migration (auto-generate)

```
alembic revision --autogenerate -m "migration message"
```

Create an empty migration

```
alembic revision -m "manual migration"
```

Apply all migrations (upgrade to latest)

```
alembic upgrade head
```

Apply migrations up to a specific revision

```
alembic upgrade <revision_id>
```

Downgrade last migration

```
alembic downgrade -1
```

Downgrade to a specific revision

```
alembic downgrade <revision_id>
```

Show migration history

```
alembic history
```

Show verbose migration history

```
alembic history --verbose
```

Stamp database with a revision (no migration run)

```
alembic stamp head
```

🗼 SQL / Database Utilities

Open database shell (PostgreSQL)

```
psql -h localhost -p 5432 -U <username> -d <database>
```

Open database shell (SQLite)

```
sqlite3 app.db
```

List tables (PostgreSQL)

```
\dt
```

Describe table (PostgreSQL)

```
\d <table_name>
```

Basic SQL checks

```sql
SELECT COUNT(*) FROM users;
SELECT * FROM users LIMIT 10;
```

Check active connections (PostgreSQL)

```
SELECT pid, usename, state, query
FROM pg_stat_activity;
```

Kill a stuck connection (PostgreSQL)

```
SELECT pg_terminate_backend(<pid>);
```

Code Quality

Run flake8

```
flake8
```

Run black formatter

```
black app tests
```

Run isort (imports)

```
isort app tests
```

Run all formatters (recommended order)

```
isort app tests && black app tests
```

Debugging & Inspection

Open interactive API docs (Swagger)

```
http://localhost:8000/docs
```

Open ReDoc

```
http://localhost:8000/redoc
```

Health check endpoint

```
curl http://localhost:8000/health
```

🚇Docker (If Used)

Build image

```
docker build -t fastapi-app .
```

Run container (dev)

```
docker run -p 8000:8000 fastapi-app
```

⚓AI / RAG (Future – Placeholder)

Run AI-related background workers

```
python -m app.workers.ai_worker
```

Run ingestion pipeline (RAG phase)

```
python -m app.ingestion.run
```

Mental Mapping (Important)

Purpose	Command
nodemon equivalent	`uvicorn --reload`
start server	`uvicorn app.main:app`

Purpose	Command
run tests	`pytest`
DB migrations	`alembic upgrade head`

Recommendation

- Keep this file at project root
- Update it when adding new tools
- Treat it as **developer contract**, not notes

This file supports **Layer 1 → Layer 9**, including future **AI & RAG** work.