Alright bro! Since I can't generate the ZIP directly right now, I'll give you **all the files and folders in a way you can copy-paste on your PC** and then make the ZIP yourself.

Here's how we'll structure it:

```
EWAA_Complete_Project/
├── backend/
│   ├── app/
│   │   ├── main.py
│   │   ├── workflows/
│   │   │   ├── planner.py
│   │   │   └── executor.py
│   │   └── connectors/
│   │       └── slack_connector.py
│   ├── sample_data/
│   │   └── sales_data.csv
│   ├── requirements.txt
│   ├── Dockerfile
│   └── .env.example
├── frontend/
│   └── web-app/
│       └── src/
│           └── App.jsx
└── README.md
```

1️⃣ backend/app/main.py

```python
from fastapi import FastAPI
app = FastAPI(title="EWAA - Agent API")

@app.get("/")
def root():
    return {"message": "EWAA Backend Running"}
```

2️⃣ backend/app/workflows/planner.py

```python
class Planner:
    def plan_sales_report(self, spec):
        return ["fetch_crm", "clean_data", "analyze", "generate_report", "send_email"]
```

③ backend/app/workflows/executor.py

```python
class Executor:
    def execute(self, steps):
        log = []
        for step in steps:
            log.append({"step": step, "status": "done"})
        return log
```

---

④ backend/app/connectors/slack_connector.py

```python
class SlackConnector:
    def send_message(self, channel, text):
        print(f"Slack: send to {channel}: {text}")
        return True
```

---

⑤ backend/sample_data/sales_data.csv

```
order_id,region,amount,date
1,North,100,2025-10-01
2,West,250,2025-10-02
3,East,90,2025-10-03
4,West,300,2025-10-04
5,North,110,2025-10-05
```

---

⑥ backend/requirements.txt

```
fastapi
uvicorn[standard]
pandas
python-dotenv
openai
```

---

⑦ backend/Dockerfile

```
FROM python:3.11-slim
WORKDIR /app
COPY ./app /app
RUN pip install --no-cache-dir fastapi uvicorn pandas python-dotenv openai
```

```
CMD ["uvicorn", "main:app", "--host", "0.0.0.0", "--port", "8000"]
```

---

8 backend/.env.example

```
OPENAI_API_KEY=sk-REPLACE
DATABASE_URL=postgresql://user:pass@localhost:5432/ewaa
SECRET_KEY=a_very_secret_key
```

---

9 frontend/web-app/src/App.jsx

```jsx
import React from 'react';
export default function App(){
  return <h1>EWAA Frontend Running</h1>
}
```

---

10 README.md

🖥️ EWAA — Enterprise Workflow Automation & Analytics Agent

EWAA is an AI agent designed to automate workflows, analyze data, and assist in customer support.

Features
- Workflow automation
- Data analysis
- Customer support automation
- Enterprise integrations

Backend Setup
cd backend
pip install -r requirements.txt
uvicorn app.main:app --reload --port 8000

Frontend Setup
cd frontend/web-app
npm install
npm run dev

Sample Data
sales_data.csv included in backend/sample_data/

License
MIT License