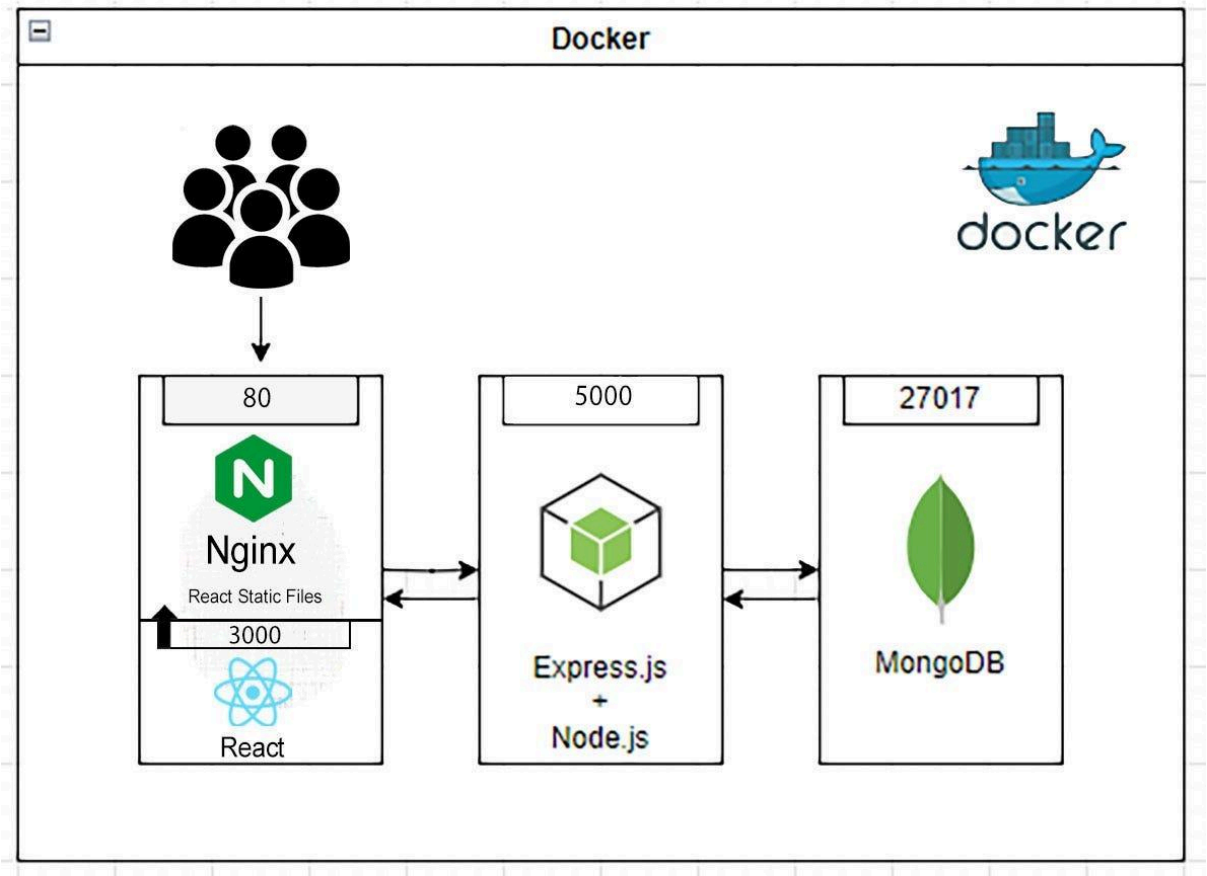
Implemented Dockerfile Support in Food Delivery Application

✅ Objective

To ensure smooth deployment and consistent environment setup, **Docker** was used to containerize the backend service of the Food Delivery App.

Docker enables:
✔ Same environment on all machines
✔ Easy deployment on cloud (AWS / Azure / Render / GCP)
✔ Faster setup for team members
✔ Isolation from system dependencies

🧱 Docker Architecture for the App



Backend (Node.js + Express) → Containerized using Docker
MongoDB → Can run via Docker or external Atlas connection

🐳 Dockerfile Used in Backend

📌 File: `Dockerfile`

```
# Step 1: Use official Node image
FROM node:18

# Step 2: Set working directory in container
WORKDIR /app

# Step 3: Copy package.json and install dependencies
COPY package*.json ./
RUN npm install

# Step 4: Copy rest of the project files
COPY . .

# Step 5: Expose backend port
EXPOSE 5000

# Step 6: Run the application
CMD ["npm", "start"]
```

▸ How to Build & Run with Docker

✅ Build a Docker Image

```
docker build -t food-app-backend .
```

✅ Run the Container

```
docker run -p 5000:5000 food-app-backend
```

➡ Backend now runs on `http://localhost:5000`

🧪 Optional: Using Docker + MongoDB Together (Docker Compose)

📌 File: `docker-compose.yml`

```
version: "3.8"
```

```yaml
services:
  backend:
    build: .
    ports:
      - "5000:5000"
    depends_on:
      - mongodb
    environment:
      - MONGO_URL=mongodb://mongodb:27017/foodapp

  mongodb:
    image: mongo
    ports:
      - "27017:27017"
```

✅ Run app + MongoDB with one command:

```
docker compose up
```

📊 Documentation Benefits

✔ Professional DevOps skill
✔ Helps in cloud deployment
✔ Shows scalable architecture
✔ Panel will be impressed in project review 😄