

# 1. إنشاء المجلد الرئيسي

```
mkdir wkalti-medical-platform  
cd wkalti-medical-platform
```

# 2. إنشاء هيكل المشروع الكامل

```
mkdir -p backend/src/  
{controllers,models,routes,middleware,servi  
ces,utils,config}  
mkdir -p frontend/web/{public,src/  
{components,pages,services,utils,styles}}  
mkdir -p mobile/{src/  
{screens,components,services,navigation,u  
tils},android,ios}  
mkdir -p database/{migrations,seeds}  
mkdir -p docs/{api,design,business}  
mkdir scripts deployment tests
```

# 3. تهيئة Git

```
git init  
echo "# وكالتي الطبية - منصة رعاية صحية ذكية" >  
README.md
```

```
# 4. إنشاء ملف .gitignore
cat > .gitignore << 'EOF'
# بيئة Node
node_modules/
npm-debug.log*
yarn-debug.log*
yarn-error.log*
```

```
# بيئة Python
__pycache__/
*.pyc
*.pyo
*.pyd
.Python
env/
venv/
.venv/
```

```
# بيئة React
build/
```

dist/  
\*.log  
.DS\_Store  
.env  
.env.local  
.env.development.local  
.env.test.local  
.env.production.local  
npm-debug.log\*  
yarn-debug.log\*  
yarn-error.log\*

# قاعدة البيانات

\*.db  
\*.sqlite  
\*.sqlite3

# IDE

.vscode/  
.idea/  
\*.swp

\*.swo

# نظام التشغيل

Thumbs.db

.DS\_Store

# Docker

.dockerignore

\*.dockerfile

docker-compose.override.yml

# الاختبارات

coverage/

.nyc\_output/

# المستندات

\*.pdf

\*.doc

\*.docx

\*.xls

\*.xlsx

EOF

```
# 5. إنشاء ملف package.json للخادم
cat > backend/package.json << 'EOF'
{
  "name": "wkalti-backend",
  "version": "1.0.0",
  "description": "Backend for Wkalti Medical
Platform",
  "main": "src/server.js",
  "scripts": {
    "start": "node src/server.js",
    "dev": "nodemon src/server.js",
    "test": "jest",
    "migrate": "node database/migrate.js",
    "seed": "node database/seed.js",
    "lint": "eslint src/",
    "docker:build": "docker build -t wkalti-
backend .",
    "docker:run": "docker run -p 3000:3000
wkalti-backend"
```

```
},  
  "keywords": ["medical", "healthcare",  
"telemedicine", "algeria"],  
  "author": "Wkalti Team",  
  "license": "MIT",  
  "dependencies": {  
    "express": "^4.18.2",  
    "sequelize": "^6.35.0",  
    "pg": "^8.11.3",  
    "pg-hstore": "^2.3.4",  
    "bcrypt": "^5.1.1",  
    "jsonwebtoken": "^9.0.2",  
    "dotenv": "^16.3.1",  
    "cors": "^2.8.5",  
    "helmet": "^7.0.0",  
    "express-validator": "^7.0.1",  
    "multer": "^1.4.5-lts.1",  
    "socket.io": "^4.7.2",  
    "twilio": "^4.19.0",  
    "nodemailer": "^6.9.7",  
    "axios": "^1.6.2",
```

```
"moment": "^2.29.4",
"winston": "^3.11.0",
"compression": "^1.7.4",
"rate-limiter-flexible": "^5.0.1",
"redis": "^4.6.11",
"swagger-ui-express": "^5.0.0",
"yamljs": "^0.3.0"
},
"devDependencies": {
  "nodemon": "^3.0.1",
  "jest": "^29.7.0",
  "supertest": "^6.3.3",
  "eslint": "^8.53.0",
  "sequelize-cli": "^6.6.2"
}
}
EOF
```

# 6. إنشاء ملف Docker للخادم

```
cat > backend/Dockerfile << 'EOF'
FROM node:18-alpine
```

WORKDIR /app

# تثبيت dependencies

COPY package\*.json ./

RUN npm ci --only=production

# نسخ الكود

COPY ..

# فتح المنفذ

EXPOSE 3000

# تشغيل التطبيق

CMD ["npm", "start"]

EOF

# 7. إنشاء docker-compose للتطوير

cat > docker-compose.dev.yml << 'EOF'

version: '3.8'



services:

postgres:

image: postgres:15-alpine

container\_name: wkalti-postgres-dev

environment:

POSTGRES\_DB: wkalti\_medical\_dev

POSTGRES\_USER: wkalti\_admin

POSTGRES\_PASSWORD:

wkalti\_pass\_123

ports:

- "5432:5432"

volumes:

- postgres\_data\_dev:/var/lib/

postgresql/data

- ./database/init.sql:/docker-entrypoint-

initdb.d/init.sql

healthcheck:

test: ["CMD-SHELL", "pg\_isready -U

wkalti\_admin"]

interval: 10s

timeout: 5s

retries: 5

redis:

image: redis:7-alpine

container\_name: wkalti-redis-dev

ports:

- "6379:6379"

volumes:

- redis\_data\_dev:/data

command: redis-server --appendonly yes

backend:

build:

context: ./backend

dockerfile: Dockerfile.dev

container\_name: wkalti-backend-dev

environment:

NODE\_ENV: development

DATABASE\_URL: postgres://

wkalti\_admin:wkalti\_pass\_123@postgres:5432/wkalti\_medical\_dev

REDIS\_URL: redis://redis:6379

JWT\_SECRET:

dev\_jwt\_secret\_key\_change\_in\_production  
ports:

- "3000:3000"

volumes:

- ./backend:/app
- /app/node\_modules

depends\_on:

postgres:

condition: service\_healthy

redis:

condition: service\_started

command: npm run dev

pgadmin:

image: dpage/pgadmin4

container\_name: wkalti-pgadmin-dev

environment:

PGADMIN\_DEFAULT\_EMAIL:

admin@wkalti.com

PGADMIN\_DEFAULT\_PASSWORD:

admin123

ports:

- "5050:80"

depends\_on:

- postgres

volumes:

postgres\_data\_dev:

redis\_data\_dev:

EOF