BIL 481 – Deployment Plan

TOBB ETÜ

University of Economics & Technology

Group Members

Helen Parlar – Artificial Intelligence Engineering

Mete Kılıç – Artificial Intelligence Engineering

Ahmet Çakmak – Medicine & Artificial Intelligence Engineering,

Table of Contents

1. Deployment Overview

1.1 Deployment Approach

The "Smart Clinical Appointment System" is deployed as a local development environment for demonstration purposes. The system utilizes a three-tier architecture:

- Frontend: React (JavaScript) - Running on localhost:3000 via dev server.
- Backend: Django REST API (Python) - Running on localhost:8000.
- Database: MySQL - Running on localhost:3306.

1.2 Tools Used

Component	Tool/Technology	Version
Frontend Server	JavaScript/React	4.x
Backend Framework	Django	5.2.8
Database	MySQL	8.0+
API Testing	Postman	Latest
Version Control	Git/GitHub	-
IDE	Visual Studio Code	Latest

2. Deployment Process

2.1 Prerequisites

Before deployment, ensure the following are installed:

- Python 3.10 or higher
- Node.js 18.x or higher
- MySQL 8.0 or higher
- Git

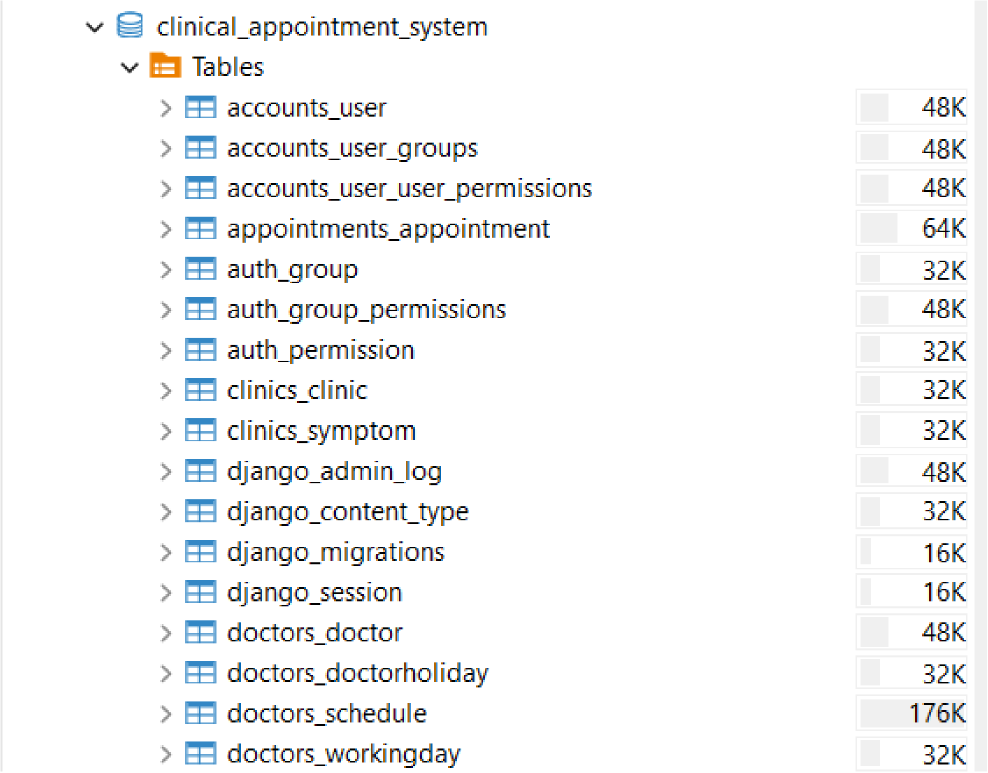
2.2 Step-by-Step Deployment

STEP 1: Clone Repository

```
git clone https://github.com/helen3331/BitTrio-ClinicalApponintmentSystem.git
cd BitTrio-ClinicalApponintmentSystem
```

STEP 2: Database Setup

1. Open MySQL Workbench or DBeaver.
2. Run the klinik_randevu_sistemi_database.sql file located in the Database/ folder to create the schema and tables.



clinical_appointment_system
Tables

accounts_user	48K
accounts_user_groups	48K
accounts_user_user_permissions	48K
appointments_appointment	64K
auth_group	32K
auth_group_permissions	48K
auth_permission	32K
clinics_clinic	32K
clinics_symptom	32K
django_admin_log	48K
django_content_type	32K
django_migrations	16K
django_session	16K
doctors_doctor	48K
doctors_doctorholiday	32K
doctors_schedule	176K
doctors_workingday	32K

STEP 3: Backend Deployment

1. **Navigate to backend directory: cd clinical_backend**
2. **Create virtual environment:**

```
python -m venv venv
```

3. **Activate virtual environment:**

```
Windows: venv\Scripts\activate
```

4. **Install dependencies: pip install -r requirements.txt**
5. **Create .env file (See Section 3 for content).**
6. **Run migrations:**

```
python manage.py makemigrations

python manage.py migrate
```

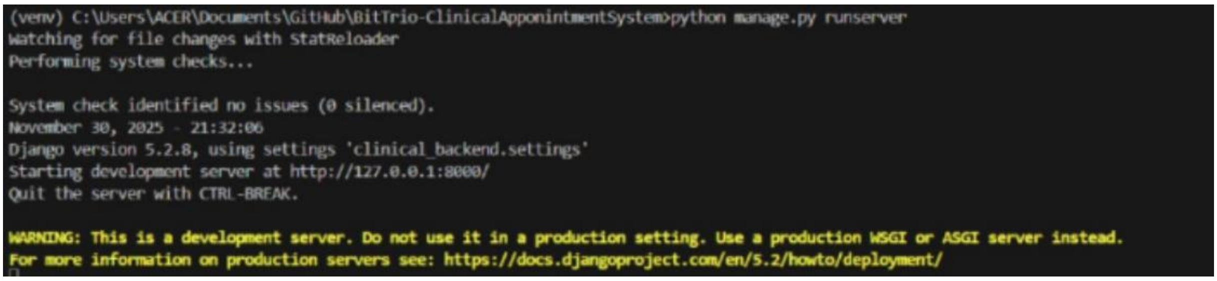
7. **Load initial data:**

```
python scripts/load_symptoms.py

python scripts/load_slots.py
```

8. **Start Django server:**

```
python manage.py runserver
```

Backend should now be running at: http://localhost:8000



```
(venv) C:\Users\ACER\Documents\GitHub\BitTrio-ClinicalApponintmentSystem>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
November 30, 2025 - 21:32:06
Django version 5.2.8, using settings 'clinical_backend.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

WARNING: This is a development server. Do not use it in a production setting. Use a production WSGI or ASGI server instead.
For more information on production servers see: https://docs.djangoproject.com/en/5.2/howto/deployment/
```

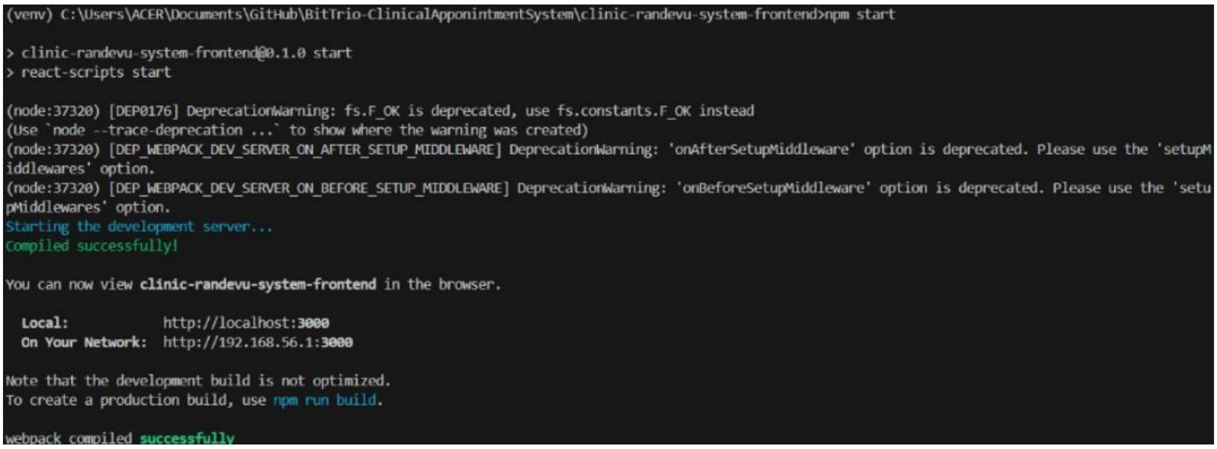
STEP 4: Frontend Deployment

1. **Open new terminal and navigate to frontend:**

```
cd clinic-randevu-system-frontend
```

2. **Install dependencies: npm install**
3. **Create .env file (See Section 3 for content).**
4. **Start development server: npm run dev**

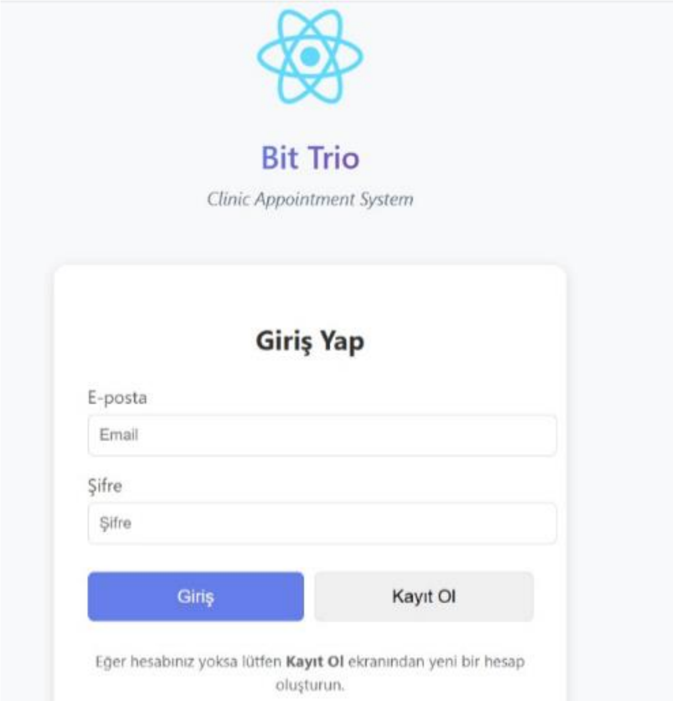
Frontend should now be running at: http://localhost:3000



```
(venv) C:\Users\ACER\Documents\GitHub\BitTrio-ClinicalApponintmentSystem\clinic-randevu-system-frontend>npm start

> clinic-randevu-system-frontend@0.1.0 start
> react-scripts start

(node:37320) [DEP0176] DeprecationWarning: fs.F_OK is deprecated, use fs.constants.F_OK instead
(Use `node --trace-deprecation ...` to show where the warning was created)
(node:37320) [DEP_WEBPACK_DEV_SERVER_ON_AFTER_SETUP_MIDDLEWARE] DeprecationWarning: 'onAfterSetupMiddleware' option is deprecated. Please use the 'setupM
iddlewares' option.
(node:37320) [DEP_WEBPACK_DEV_SERVER_ON_BEFORE_SETUP_MIDDLEWARE] DeprecationWarning: 'onBeforeSetupMiddleware' option is deprecated. Please use the 'setu
pMiddlewares' option.
Starting the development server...
Compiled successfully!

You can now view clinic-randevu-system-frontend in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.56.1:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

STEP 5: Verification

1. Open browser and navigate to: http://localhost:3000
2. Test registration flow.
3. Verify API connection by checking Network tab in browser developer tools.



Bit Trio

Clinic Appointment System

Giriş Yap

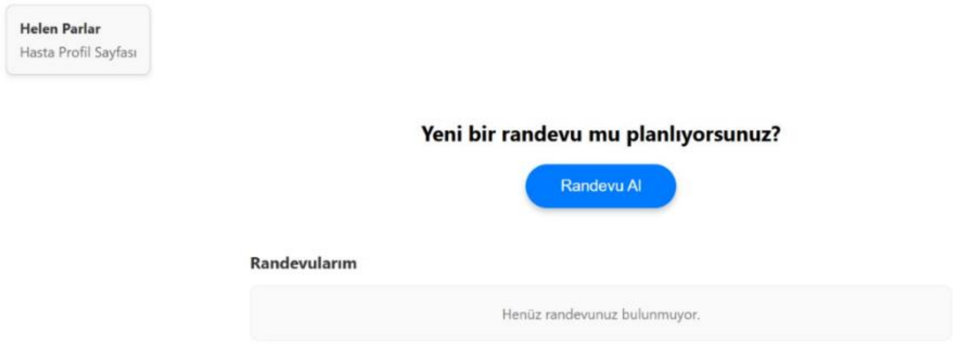
E-posta

Email

Şifre

Şifre

Eğer hesabınız yoksa lütfen **Kayıt Ol** ekranından yeni bir hesap oluşturun.


Helen Parlar
Hasta Profil Sayfası

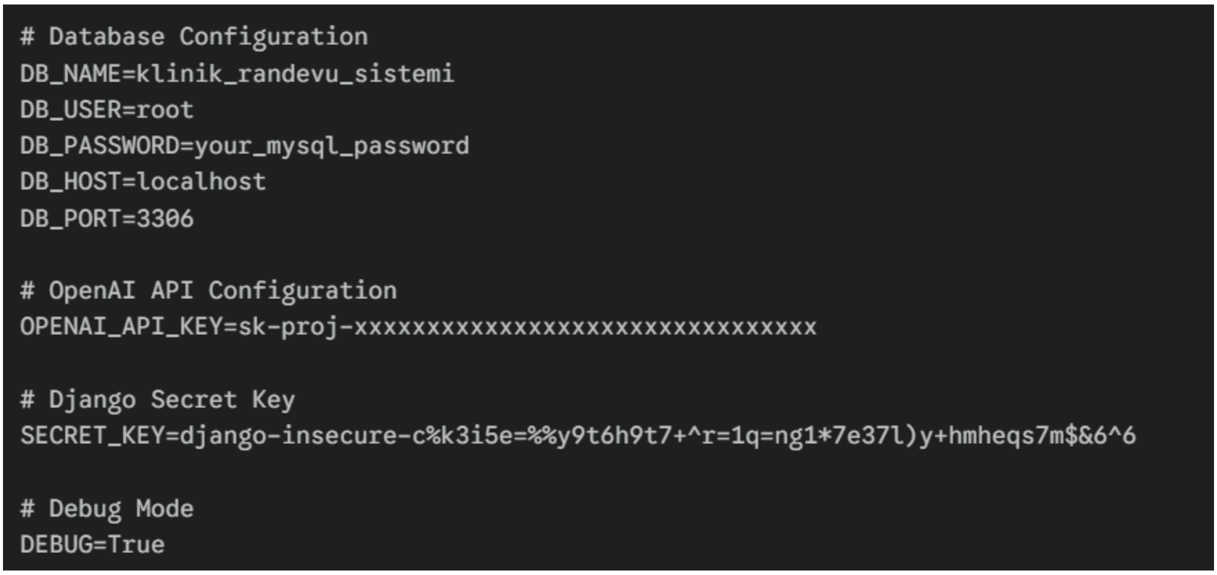
Yeni bir randevu mu planlıyorsunuz?

Randevularım

Henüz randevunuz bulunmuyor.

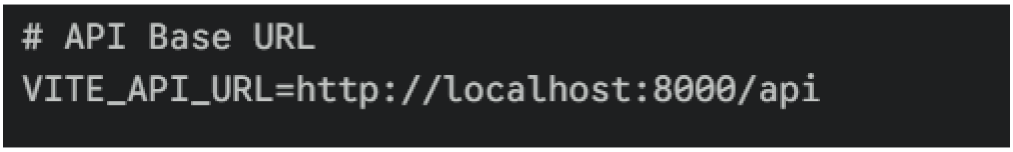
3. Configuration Plan

3.1 Backend Configuration



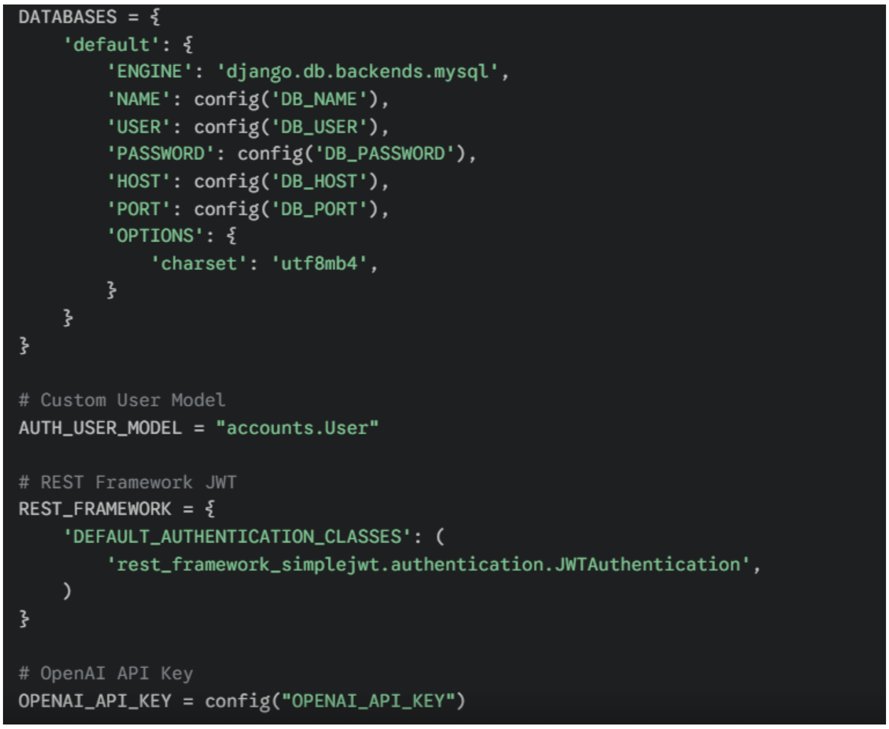
```
# Database Configuration
DB_NAME=klinik_randevu_sistemi
DB_USER=root
DB_PASSWORD=your_mysql_password
DB_HOST=localhost
DB_PORT=3306

# OpenAI API Configuration
OPENAI_API_KEY=sk-proj-xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

# Django Secret Key
SECRET_KEY=django-insecure-c%k3i5e=%%y9t6h9t7+^r=1q=ng1*7e37l)y+hmheqs7m$&6^6

# Debug Mode
DEBUG=True
```

3.2 Frontend Configuration



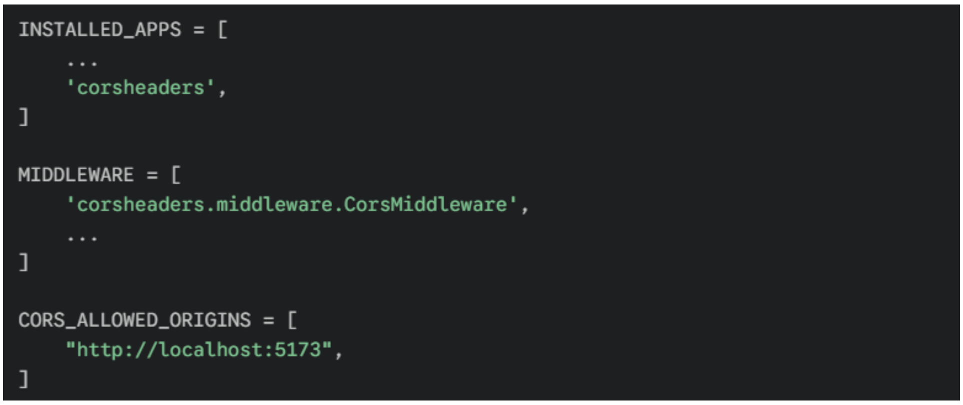
```
# API Base URL
VITE_API_URL=http://localhost:8000/api
```

3.3 Django Settings Configuration Key configurations in clinical_backend/settings.py:



```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': config('DB_NAME'),
        'USER': config('DB_USER'),
        'PASSWORD': config('DB_PASSWORD'),
        'HOST': config('DB_HOST'),
        'PORT': config('DB_PORT'),
        'OPTIONS': {
            'charset': 'utf8mb4',
        }
    }
}

# Custom User Model
AUTH_USER_MODEL = "accounts.User"

# REST Framework JWT
REST_FRAMEWORK = {
    'DEFAULT_AUTHENTICATION_CLASSES': (
        'rest_framework_simplejwt.authentication.JWTAuthentication',
    )
}

# OpenAI API Key
OPENAI_API_KEY = config("OPENAI_API_KEY")
```

3.4 CORS Configuration Added to settings.py to allow frontend connection:



```
INSTALLED_APPS = [
    ...
    'corsheaders',
]

MIDDLEWARE = [
    'corsheaders.middleware.CorsMiddleware',
    ...
]

CORS_ALLOWED_ORIGINS = [
    "http://localhost:5173",
]
```

3.5 Final Configuration Checklist

Before demo, verify:

- MySQL database is running on port 3306.
- .env files are configured correctly.
- Backend server is running on port 8000.
- Frontend server is running on port 3000.
- OpenAI API key is valid.
- Initial data (symptoms, slots) is loaded.

4. Task Matrix

Task	**Responsible Member(s)**
Database Setup & SQL Schema	**Mete Kılıç**
Backend Deployment Steps	**Helen Parlar,**
Frontend Deployment Steps	**Ahmet Çakmak**
Configuration Files (.env)	**All Members**
Verification & Testing	**All Members**
Documentation & Screenshots	**All Members**
Final Review	**All Members**