



**Универзитет “Св. Кирил и Методиј”**  
**Факултет за информатички науки и компјутерско**  
**инженерство**



## **ПРОЕКТНА ЗАДАЧА**

**По предметот**

**КОНТИНУИРАНА ИНТЕГРАЦИЈА И ИСПОРАКА**

**Тема**

**EventsApp**

**Ментори:**

**Проф. Д-р. Милош Јовановиќ**

**Проф. Д-р. Панче Рибарски**

**Изработил:**

**Кире Атанасков**

*Скопје, 2024*

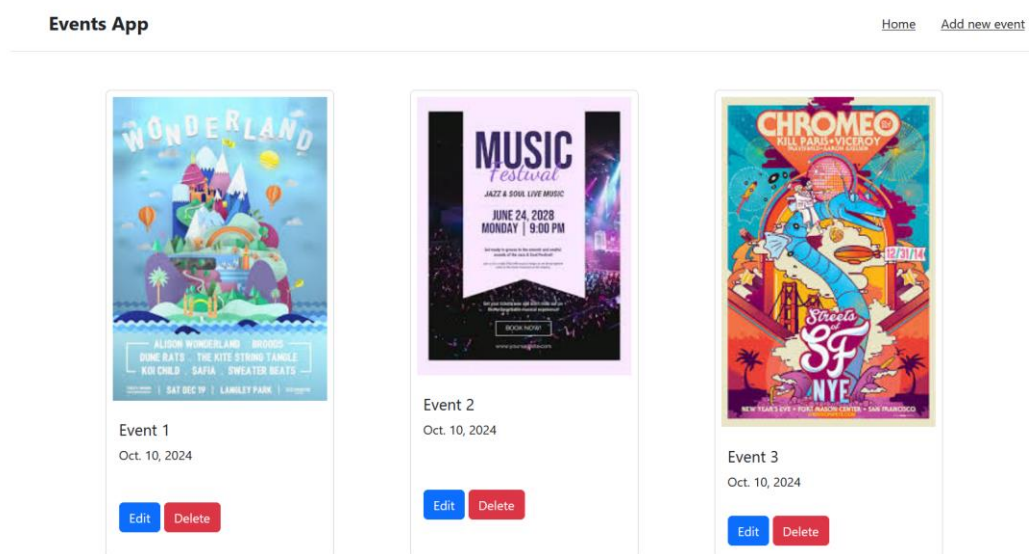
Проектот вклучува CRUD апликација за менаџирање на настани развиена користејќи Django и PostgreSQL. Апликацијата е контејнизирана користејќи Docker и е извршена нејзина испорака со Kubernetes.

Линкови до GitHub репозиториумот и Dockerhub сликата за апликацијата.

GitHub: <https://github.com/kireatanaskov/Django-KIII>

Dockerhub: <https://hub.docker.com/repository/docker/kireatanaskov/kiii-project/general>

Изглед на почетната страница на апликацијата



Форма за додавање/менување на настан

## Add event

Name:

Date:

Poster:

No file selected.

Open air: ☐

**Dockerfile** за апликацијата кој ги копира фајловите од локалниот фолдер во контејнерот, ги инсталира потребните библиотеки со помош на `pip`, ја поставува портата на која што ќе слуша контејнерот и се извршуваат команди за правење на миграции во базата и стартување на апликацијата.

```
FROM python:3.12.3-slim

WORKDIR /app

COPY ./app /app

RUN pip install --no-cache-dir -r requirements.txt

EXPOSE 80

ENV PYTHONUNBUFFERED=1

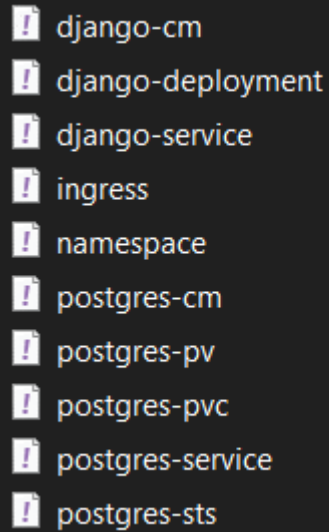
CMD ["sh", "-c", "python manage.py migrate && python manage.py runserver 0.0.0.0:80"]
```

**Docker-compose** фајл за стартување на сервисот за апликацијата и базата на податоци.

```
version: '3.8'

services:
  web:
    build: .
    command: sh -c "python manage.py migrate && python manage.py runserver 0.0.0.0:80"
    volumes:
      - ./app:/app
    ports:
      - "80:80"
    depends_on:
      - db
    environment:
      - DB_HOST=db
      - DB_NAME=eventsApp
      - DB_USER=postgres
      - DB_PASSWORD=postgres
      - DB_PORT=5432
  db:
    image: postgres:latest
    ports:
      - "5432:5432"
    environment:
      - POSTGRES_DB=eventsApp
      - POSTGRES_USER=postgres
      - POSTGRES_PASSWORD=postgres
```

Манифести за успешно испорачување на апликацијата со Kubernetes



- django-cm
- django-deployment
- django-service
- ingress
- namespace
- postgres-cm
- postgres-pv
- postgres-pvc
- postgres-service
- postgres-sts

namespace.yml

```
apiVersion: v1
kind: Namespace
metadata:
  name: django-app
```

postgres-cm.yml

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: postgres-secret
  namespace: django-app
  labels:
    app: postgres
data:
  POSTGRES_DB: eventsApp
  POSTGRES_USER: postgres
  POSTGRES_PASSWORD: postgres
```

postgres-pv.yml

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: postgres-volume
  namespace: django-app
  labels:
    type: local
    app: postgres
spec:
  storageClassName: manual
  capacity:
    storage: 10Gi
  accessModes:
    - ReadWriteMany
  hostPath:
    path: /data/postgresql
```

postgres-pvc.yml

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: postgres-pvc
  namespace: django-app
  labels:
    app: postgres
spec:
  storageClassName: manual
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 10Gi
```

## postgres-service.yml

```
apiVersion: v1
kind: Service
metadata:
  name: postgres-service
  namespace: django-app
  labels:
    app: postgres
spec:
  ports:
    - port: 5432
      targetPort: 5432
  selector:
    app: postgres
```

## postgres-sts.yml

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: postgres
  namespace: django-app
spec:
  serviceName: postgres-service
  replicas: 1
  selector:
    matchLabels:
      app: postgres
  template:
    metadata:
      labels:
        app: postgres
    spec:
      containers:
        - name: postgres
          image: 'postgres:16.3'
          imagePullPolicy: IfNotPresent
          ports:
            - containerPort: 5432
          envFrom:
            - configMapRef:
                name: postgres-secret
          volumeMounts:
            - mountPath: /var/lib/postgresql/data
              name: postgresdata
      volumeClaimTemplates:
        - metadata:
            name: postgresdata
          spec:
            accessModes: ["ReadWriteOnce"]
            resources:
              requests:
                storage: 10Gi
```

## django-cm.yml

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: django-cm
  namespace: django-app
data:
  DB_HOST: "postgres-service"
  DB_USER: "postgres"
  DB_PASSWORD: "postgres"
  DB_NAME: "eventsApp"
  DB_PORT: "5432"
```

## django-deployment.yml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: django-app-deploy
  namespace: django-app
spec:
  replicas: 3
  selector:
    matchLabels:
      app: django-application
  template:
    metadata:
      labels:
        app: django-application
    spec:
      containers:
        - image: kireatanaskov/kiii-project:latest
          imagePullPolicy: Always
          name: django-app-container
          envFrom:
            - configMapRef:
                name: django-cm
          ports:
            - containerPort: 80
```

django-service.yml

```
apiVersion: v1
kind: Service
metadata:
  name: django-service
  namespace: django-app
spec:
  selector:
    app: django-application
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```