



SPRINT 2 – FULL & FINAL (REACT + DJANGO)



OVERALL ARCHITECTURE

```
React (Frontend)
  ↓ HTTP (fetch)
Django REST APIs
  ↓
MySQL (RAW SQL)
```



BACKEND SETUP (DJANGO + MYSQL)

1 Create Backend Project

```
django-admin startproject backend
cd backend
python manage.py startapp authapp
python manage.py startapp projectapp
python manage.py startapp sprintapp
```

2 Install Dependencies

```
pip install django django-cors-headers mysqlclient
```

3 settings.py (FINAL)

```
INSTALLED_APPS = [
    'corsheaders',
    'authapp',
    'projectapp',
    'sprintapp',

    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
]
```

```

MIDDLEWARE = [
    'corsheaders.middleware.CorsMiddleware',
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
]

CORS_ALLOW_ALL_ORIGINS = True

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'project_catalyst',
        'USER': 'root',
        'PASSWORD': 'your_mysql_password',
        'HOST': 'localhost',
        'PORT': '3306',
    }
}

```

4 backend/urls.py

```

from django.urls import path, include

urlpatterns = [
    path('api/', include('authapp.urls')),
    path('api/', include('projectapp.urls')),
    path('api/', include('sprintapp.urls')),
]

```

5 Database (RUN ONCE)

```

CREATE DATABASE project_catalyst;
USE project_catalyst;

CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    email VARCHAR(100),
    password VARCHAR(100)
);

CREATE TABLE projects (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100),
    owner_id INT
);

CREATE TABLE sprints (
    id INT AUTO_INCREMENT PRIMARY KEY,
    project_id INT,
    name VARCHAR(100),
    start_date DATE,
    end_date DATE,
    status VARCHAR(20)
);

```



BACKEND APPS (FINAL CODE)

projectapp/urls.py

```
from django.urls import path
from .views import create_project, get_projects

urlpatterns = [
    path('projects/create/', create_project),
    path('projects/', get_projects),
]
```

projectapp/views.py

```
from django.http import JsonResponse
from django.views.decorators.csrf import csrf_exempt
from django.db import connection
import json

@csrf_exempt
def create_project(request):
    data = json.loads(request.body)
    with connection.cursor() as cursor:
        cursor.execute(
            "INSERT INTO projects (name, owner_id) VALUES (%s, %s)",
            [data['name'], data['owner_id']])
    return JsonResponse({"message": "Project Created"})

def get_projects(request):
    owner_id = request.GET.get("owner_id")
    with connection.cursor() as cursor:
        cursor.execute(
            "SELECT id, name FROM projects WHERE owner_id=%s",
            [owner_id])
    rows = cursor.fetchall()

    return JsonResponse(
        [{"id": r[0], "name": r[1]} for r in rows],
        safe=False
    )
```

sprintapp/urls.py

```
from django.urls import path
from .views import create_sprint, get_sprints

urlpatterns = [
    path('sprints/create/', create_sprint),
    path('sprints/<int:project_id>/', get_sprints),
]
```

sprintapp/views.py

```
from django.http import JsonResponse
from django.views.decorators.csrf import csrf_exempt
from django.db import connection
import json

@csrf_exempt
def create_sprint(request):
    data = json.loads(request.body)
    with connection.cursor() as cursor:
        cursor.execute("""
            INSERT INTO sprints
            (project_id, name, start_date, end_date, status)
            VALUES (%s,%s,%s,%s,'Active')
        """, [
            data['project_id'],
            data['name'],
            data['start_date'],
            data['end_date']
        ])
    return JsonResponse({"message": "Sprint Created"})

def get_sprints(request, project_id):
    with connection.cursor() as cursor:
        cursor.execute(
            "SELECT id, name FROM sprints WHERE project_id=%s",
            [project_id]
        )
    rows = cursor.fetchall()

    return JsonResponse(
        [{"id": r[0], "name": r[1]} for r in rows],
        safe=False
    )
```

▶ Run Backend

python manage.py runserver

Backend runs at:

👉 <http://127.0.0.1:8000/>



FRONTEND SETUP (REACT)

1 Create React App

```
npx create-react-app frontend
cd frontend
npm start
```

2 Install Add-ons

```
npm install axios react-router-dom
```

3 src/api.js

```
const API = "http://127.0.0.1:8000/api";

export const createProject = (data) =>
  fetch(` ${API}/projects/create/`, {
    method: "POST",
    headers: { "Content-Type": "application/json" },
    body: JSON.stringify(data),
  });

export const getProjects = (owner_id) =>
  fetch(` ${API}/projects/?owner_id=${owner_id}`);

export const createSprint = (data) =>
  fetch(` ${API}/sprints/create/`, {
    method: "POST",
    headers: { "Content-Type": "application/json" },
    body: JSON.stringify(data),
  });

export const getSprints = (project_id) =>
  fetch(` ${API}/sprints/${project_id}/`);
```

4 Project Page (WEB FORM)

src/Projects.js

```
import { createProject, getProjects } from "./api";
import { useEffect, useState } from "react";

export default function Projects() {
  const [projects, setProjects] = useState([]);

  const loadProjects = async () => {
    const res = await getProjects(1);
    setProjects(await res.json());
  };

  const submit = async () => {
    await createProject({ name: "My First Project", owner_id: 1 });
    loadProjects();
  };

  useEffect(() => {
    loadProjects();
  }, []);
}
```

```

        return (
      <>
        <h2>Projects</h2>
        <button onClick={submit}>Create Project</button>
        <ul>
          {projects.map(p => <li key={p.id}>{p.name}</li>) }
        </ul>
      </>
    ) ;
}

```

5 Sprint Page (WEB FORM)

src/Sprints.js

```

import { createSprint, getSprints } from "./api";
import { useEffect, useState } from "react";

export default function Sprints() {
  const [sprints, setSprints] = useState([]);

  const loadSprints = async () => {
    const res = await getSprints(1);
    setSprints(await res.json());
  };

  const submit = async () => {
    await createSprint({
      project_id: 1,
      name: "Sprint 1",
      start_date: "2025-01-10",
      end_date: "2025-01-17"
    });
    loadSprints();
  };

  useEffect(() => {
    loadSprints();
  }, []);

  return (
    <>
      <h2>Sprints</h2>
      <button onClick={submit}>Create Sprint</button>
      <ul>
        {sprints.map(s => <li key={s.id}>{s.name}</li>) }
      </ul>
    </>
  );
}

```

6 App.js (LINK-BASED NAVIGATION)

```
import Projects from "./Projects";
```

```
import Sprints from "./Sprints";  
  
export default function App() {  
  return (  
    <>  
    <Projects />  
    <hr />  
    <Sprints />  
  </>  
);  
}
```



HOW YOU USE IT (NO POSTMAN)

1 Open browser

👉 <http://localhost:3000>

2 Click **Create Project**

✓ Project saved in MySQL

3 Click **Create Sprint**

✓ Sprint linked to project

4 Data loads automatically

✓ Full web-based flow