# Phase 1 – Conception Phase

## Title: BürgerNetz - (Connecting citizens, shaping the city)


## Abstract:

BurgerNetz is a web platform where the citizens of the city can report problems in the city to the government. People can report things like potholes, broken streetlights, garbage issues or any other problems. The platform also lets them track the status of their reports so they know when the issue is fixed.

The system shows reports on a city map, helping everyone see where problems are most common. It also organizes issues automatically, so the city administration can prioritize and solve them faster. An analytics dashboard shows trends and areas that need attention.

BugerNetz is built using Angular for the frontend, python/FastAPI for the backend, and PostgreSQL for the database. It uses cloud storage and modern web tools, making it fast and scalable. Combining citizens reporting with useful insights, the project makes cities smarter, more responsive and more transparent which gives a citizens stronger voice in improving their community.


## Goals:

1. The platform will allow citizens to easily report city issues such as potholes, broken streetlights and garbage collection problems.
2. Citizens will be able to track the status of their reports.
3. The system will visualize reported issues on a map to highlight problem hotspots and provide useful information to every citizen.
4. Government will have access to an analytics dashboard showing trends, response times and priority areas for repair.
5. BürgerNetz will encourage civic engagement by giving citizens an active role in improving their city.
6. The system will be designed to be scalable and accessible on multiple devices, allowing usability for a large number of citizens.

## Tech Stack:

- **Frontend Framework:** Angular
- **UI Library:** Angular Material
- **Mapping:** Leaflet.js / Mapbox
- **Data Visualization:** Chart.js / D3.js
- **Backend Framework:** Python FastAPI
- **Database:** PostgreSQL
- **Geospatial Queries:** PostGIS
- **Frontend Hosting:** Vercel / AWS Amplify
- **Backend Hosting:** AWS EC2 / Render / Railway
- **Database Hosting:** AWS RDS (PostgreSQL)
- **Version Control:** Git + GitHub