

PollutionCheck App Case Study

1	Name of the Project	PollutionCheck App
2	Objective/ Vision	<p>Build a system to keep track of Air Quality of different cities. A user should be able to search a city Air quality index. Users should be able to register themselves in the application and save search data as a logged user. Logged in user should be able to put cities in watch list for their air quality index.</p> <p>The application needs to fetch data from https://www.airvisual.com/air-pollution-data-api using which watch List of Cities can be created</p>
3	Users of the System	All Internet users
4	Functional Requirements	<ol style="list-style-type: none"> 1) Home Page should consist of Register page through which a user can register himself. Upon registration, the user should be able to login into his account. 2) User home page should also have options to edit the profile and change his/her password. 3) Air Watcher - View to select the country and display the States 4) View to select the State to show the Air Quality of all the Cities of that state. 5) There should be a functionality for putting the cities in watch list and manage the watchlist
5	Non-functional requirements	<ol style="list-style-type: none"> a) App should be accessible from any location with access to the Internet. b) App should be responsive to display consistently across multiple device screens. c) App should have an intuitive UI that can be operated by novice-expert Internet users
6	Tools and Technologies to be used	<ol style="list-style-type: none"> 1. VCS : Gitlab 2. Middleware : Spring Boot 3. Frond end : Angular/React 4. Data Store : MongoDB / MySQL 5. Testing : JUnit, Mocha, Chai, Jest, Protractor 6. Container : Docker 7. Bug Fix : Sonarlint 8. CI : Gitlab

User Stories

1	As a user I should be able to register with the application so that I can login and use the functionalities of the application.
2	As a user, I should be able to login with my user name and password in order to access the functionalities of the application.
3	As a user, I should be able to login with my Gmail account to access the functionalities of the application. (optional requirement)
4	As a user I should be able to search resources to view their details
5	As a user, I should be able to save resources to a watchlist/favourite so that I can access them later
6	As a user, I should be able to access items saved to my watchlist /favourite

Notes:

- The application should be based on microservices architecture
- API Gateway pattern should be implemented using Spring Cloud Gateway
- Services should register themselves with Eureka Service Discovery server.
- All layers of microservices should be covered with automated unit and integration tests
- All microservice endpoints should have API documentation

High Level Architecture Diagram

