**Project Title: Nearby Bank Finder – Microservice Using Spring Boot and Google Maps API**

**Technology Stack:**

| **Technology** | **Purpose** |
| --- | --- |

|  |  |
| --- | --- |
| Java (Spring Boot) | Backend REST API development |

|  |  |
| --- | --- |
| Google Geocoding API | Convert ZIP code to latitude and longitude |

|  |  |
| --- | --- |
| Google Places API | Fetch nearby banks based on location |

|  |  |
| --- | --- |
| REST API | Expose endpoints for client interaction |

|  |  |
| --- | --- |
| IntelliJ IDEA | IDE used for development and testing |

|  |  |
| --- | --- |
| Postman | Used for manual API testing |

|  |  |
| --- | --- |
| GitHub | Version control and project tracking |

**Overview:**

This project is a backend microservice application built using Spring Boot. It accepts a ZIP code as input and responds with a list of nearby banks using real-time data from Google APIs.

**Endpoint:**

| **HTTP Method** | **Endpoint** | **Description** |
| --- | --- | --- |

|  |  |  |
| --- | --- | --- |
| GET | /banks/nearby?zip=71270 | Returns banks within 10-mile radius |
|  |  |  |

**Workflow:**

**User Input**  
A client sends a GET request with a ZIP code as a query parameter.

 **Geolocation**  
The app uses **Google Geocoding API** to convert the ZIP code into latitude and longitude.

 **Nearby Search**  
It then uses **Google Places API** to search for banks near those coordinates, within a 10-mile radius (16,093 meters).

 **Filtering and Formatting**  
The system filters and formats the bank data into a list that includes:

* Bank name
* Address
* Distance in miles (rounded to 1 decimal)

 **Response Output**  
The application returns this list as a JSON response.

Sample Request: GET http://localhost:8080/banks/nearby?zip=71270

**Sample response:**

[

{

"name": "Cadence Bank",

"address": "300 North Trenton Street, Ruston",

"distanceInMiles": 4.7

},

{

"name": "Chase Bank",

"address": "297 North Service Road East, Ruston",

"distanceInMiles": 5.5

}

]

**Code Structure:**

**com.nearbybank.api**

**├── controller → BankController.java**

**├── service → GoogleService.java**

**├── model → BankInfo.java**

**└── ApiApplication.java → Main class**

**Security and Validation**

* Only valid ZIP codes are processed
* If a ZIP code is invalid or returns no results, the system returns an empty list with no errors
* All external API keys are securely stored in application.properties

**Testing**

* The application was tested using **Postman** and the built-in **REST client in IntelliJ**
* Logs were added to monitor Geocoding and Places API responses
* Output has been validated for ZIP codes with active bank listings

**Conclusion :**

This Spring Boot microservice is a modular, clean, and real-time solution to fetch bank information based on ZIP code. It follows REST principles and can be integrated with mobile apps, websites, or CRM systems. The application is built with clarity, maintainability, and scalability in mind.