**README AUTO GENERATOR AND UPDATER**

**✅ Project Objective:**

Build a Java application that **reads a GitHub project**, analyzes its **structure, files, and metadata**, and **automatically generates or updates the README.md** file with standard sections like:

* Project title
* Description
* Features
* Installation
* Usage
* Technologies used
* License

**📂 Project Modules:**

1. **GitHub Repository Reader** – Reads project metadata and directory structure.
2. **README Generator** – Creates/updates markdown content.
3. **File I/O Manager** – Handles reading/writing of README.md.
4. **Command-Line Interface (CLI)** – Lets users interact with the tool.

**🧑‍💻 Technologies Used:**

* Java 11+
* GitHub API (REST)
* JSON (via Jackson or org.json)
* Markdown (basic string generation)
* Java I/O and File handling

**🧾 Sample Code (Simplified Version)**

Below is a simple version of the code that:

* Takes project name and description as input
* Generates or updates a README.md file

CODE:

import java.io. \*;

import java.util.Scanner;

public class ReadmeUpdater {

private static final String README\_PATH = "README.md";

public static void main (String [] args) {

Scanner scanner = new Scanner (System.in);

System.out.println("=== README.md Generator ===");

System.out.print("Enter Project Title: ");

String title = scanner.nextLine();

System.out.print("Enter Project Description: ");

String description = scanner.nextLine();

System.out.print("Enter Features (comma separated): ");

String features = scanner.nextLine();

System.out.print("Enter Technologies Used (comma separated): ");

String technologies = scanner.nextLine();

String content = generateReadmeContent(title, description, features, technologies);

try {

writeToFile(README\_PATH, content);

System.out.println("README.md updated successfully.");

} catch (IOException e) {

System.err.println("Error writing README.md: " + e.getMessage());

}

scanner.close();

}

private static String generateReadmeContent(String title, String description, String features, String technologies) {

return """

# %s

## Description

%s

## Features

%s

## Technologies Used

%s

## Installation

```bash

git clone <your-repo-url>

cd <project-directory>

```

## Usage

Run the application using your preferred IDE or terminal.

## License

This project is licensed under the MIT License.

""“. formatted (title, description, formatList(features), formatList(technologies));

}

private static String formatList (String input) {

String [] items = input.split(",");

StringBuilder formatted = new StringBuilder ();

for (String item: items) {

formatted.append("- "). append(item.trim()). append("\n");

}

return formatted.toString();

}

private static void writeToFile (String filePath, String content) throws IOException {

BufferedWriter writer = new Buffered Writer(new FileWriter(filePath));

writer.write(content);

writer.close();

}

}

**💡 Project Extensions:**

* Auto-fetch GitHub metadata using the GitHub API (project name, contributors, last commit, etc.).
* Integrate with GUI (Swing or JavaFX).
* Support markdown templates.
* Add language-specific badges (Java, Maven, GitHub Actions, etc.).

**📁 Deliverables for Submission:**

* Full Java source code
* Sample README.md before and after generation
* Report explaining code and structure
* Screenshots of usage
* GitHub repo (optional)