**Name: Neha Gade**

**Roll No: 16010122325 C-3**

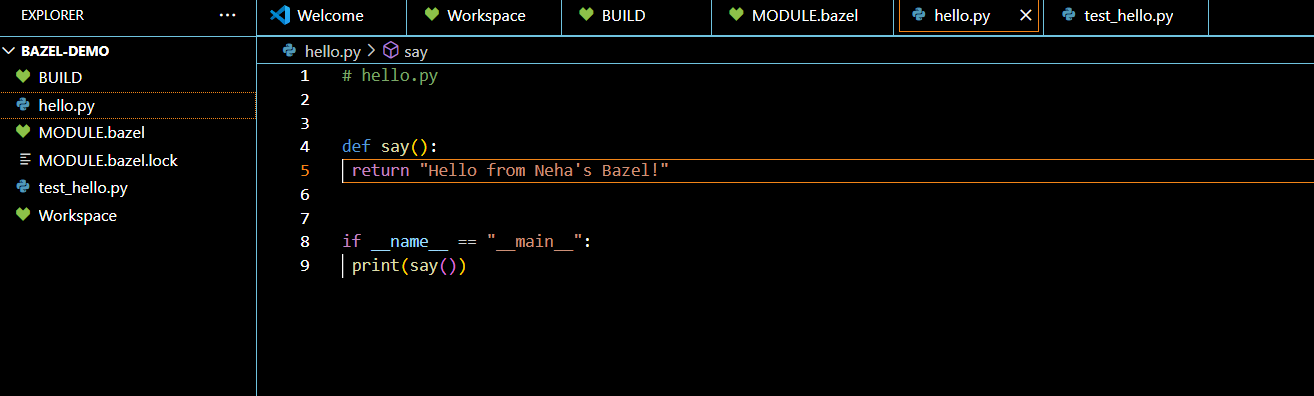
**Devops Doc IA1  
Tool-Bazel by Google**

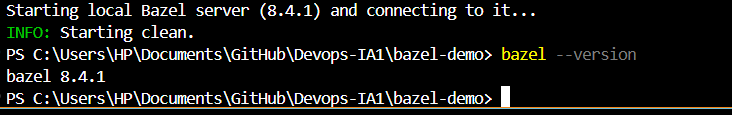
**Bazel (local implementation)** — Bazel is a fast, open-source build and test automation tool created by Google for large-scale, multi-language projects. It supports incremental and reproducible builds, making it highly efficient for complex monorepos. As an alternative to Maven and Gradle, Bazel is optimized for speed, scalability, and CI/CD integration.

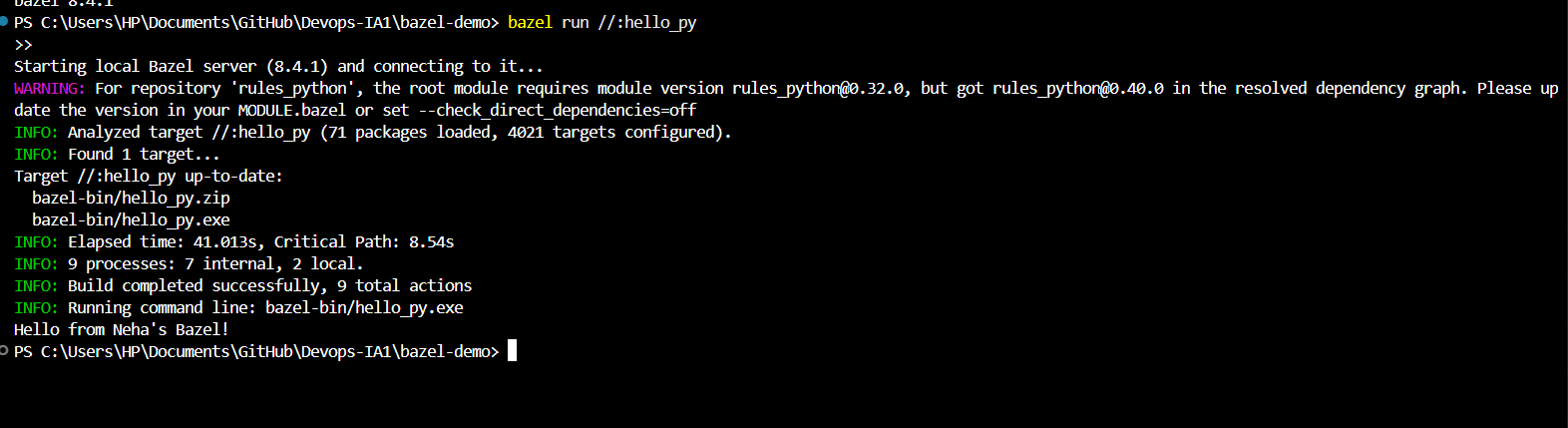
 **Scope:** Local (Windows) installation and demonstration of Bazel building, running, and testing a small sample project (Python example for simplicity).

 **Objective:** Show end-to-end steps so you can record the working demo, include code/config files in the repo, and write the report with screenshots.

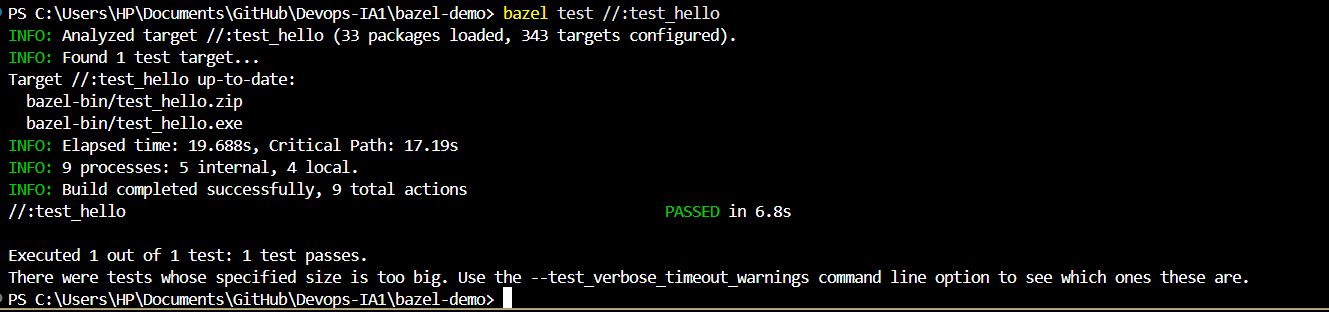
IMPLEMENTATION :

1.building bazel  


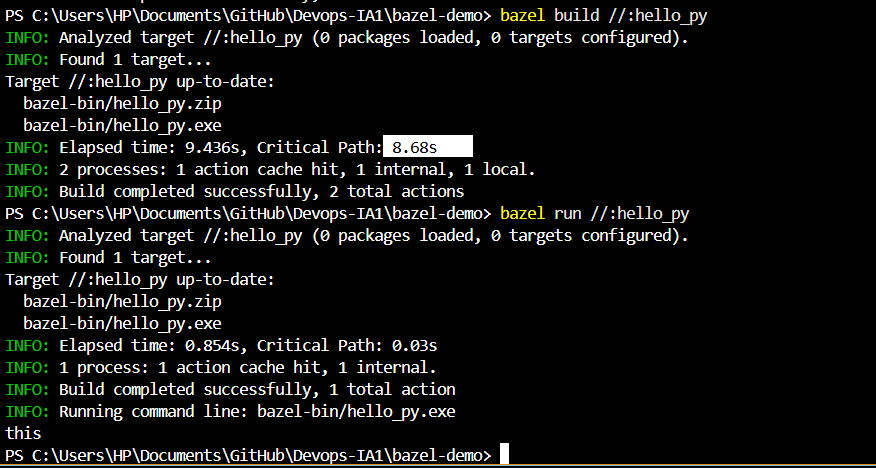
2.bazel version-  
  
  
3.running scipts

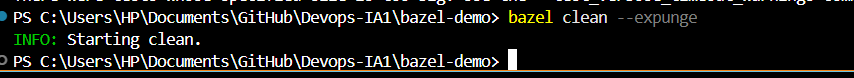


4.test



5.Incremental build: terminal showing times before/after modifying source.



6. clear screen  


2)

**Comparative analysis (Bazel vs Gradle / Maven) — short for report**

**1. Ease of setup & use**

* **Bazel:** Steeper learning curve (Starlark rules, WORKSPACE/BUILD model).
* **Gradle:** Easier for Java/Kotlin projects (Groovy/Kotlin DSL) and very familiar to many devs.
* **Maven:** Convention-over-configuration; easier for standard Java projects but verbose POMs.

**2. Performance & Scalability**

* **Bazel:** Best for very large monorepos — incremental builds + caching + parallelism.
* **Gradle/Maven:** Good for normal projects; Gradle has incremental features but may be slower on very large repos.

**3. CI/CD Integration**

* **Bazel:** Integrates well with CI systems; reproducible builds are great for CI.
* **Gradle/Maven:** Native support in many CI providers and wide plugin ecosystems.

**4. Advantages & Limitations**

* **Bazel Advantages:** Speed, hermetic reproducible builds, multi-language monorepo support.
* **Bazel Limitations:** Smaller ecosystem, steeper learning curve, more complex to add custom rules.

**Advantages of Bazel?**

* **Speed:** It caches build results and rebuilds only what’s needed.
* **Correctness:** Builds are hermetic and reproducible.
* **Scalability:** Handles very large, complex projects.
* **Cross-language:** Supports many programming languages in the same build.

**Challenges & troubleshooting (to include in report)**

* Windows path & permission issues (ensure bazel.exe is in PATH).
* Toolchain downloads blocked by firewall or antivirus — allow bazel/bazelisk.
* Missing JDK or mismatch of JAVA\_HOME — set properly.
* If using Python examples: ensure python is on PATH and compatible version is used.

**Example use cases:**

* Google uses Bazel to build massive codebases for Android, Search, and more.
* Companies with large codebases and many dependencies use Bazel to speed up their build pipelines.
* Open source projects that want to support multiple languages and platforms.