Name – Tanishq Thuse

Roll no – 52

PRN – 12310237

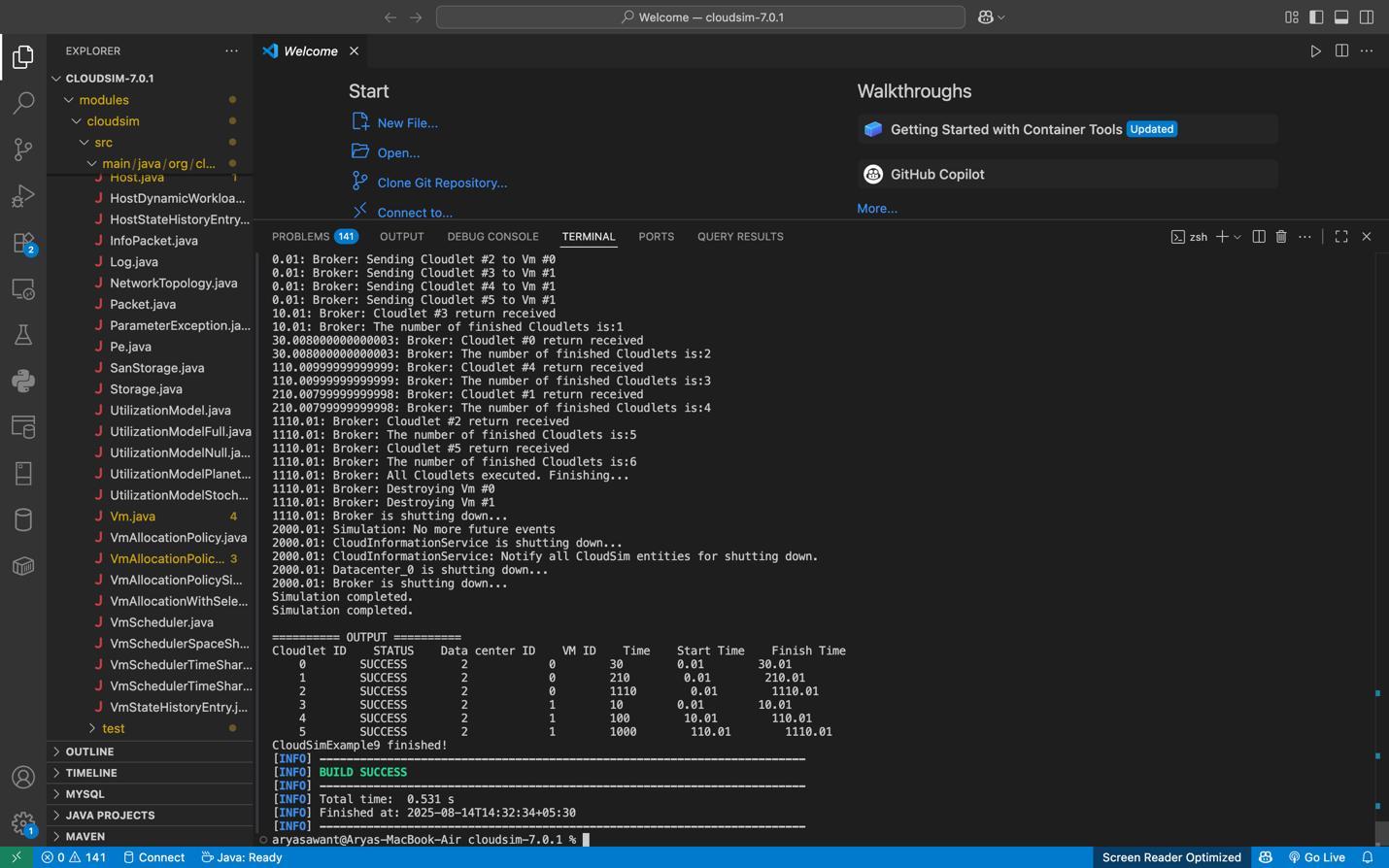
Subject - Cloud Computing

**Assignment 5: CloudSim Installation and Setup Guide**

# 1. Install Java JDK (Recommended: JDK 21)

CloudSim requires Java to run. We recommend installing Java Development Kit (JDK) version 21.  
- Download JDK 21:  
 • Oracle JDK: https://www.oracle.com/java/technologies/javase/jdk21-archive-downloads.html  
 • Adoptium OpenJDK: https://adoptium.net/releases.html?variant=openjdk21  
- Installation steps:  
 1. Download the Windows x64 Installer `.msi` file from either Oracle or Adoptium.  
 2. Run the installer and follow the prompts.  
 3. By default, the JDK installs to a folder like C:\Program Files\Java\jdk-21.

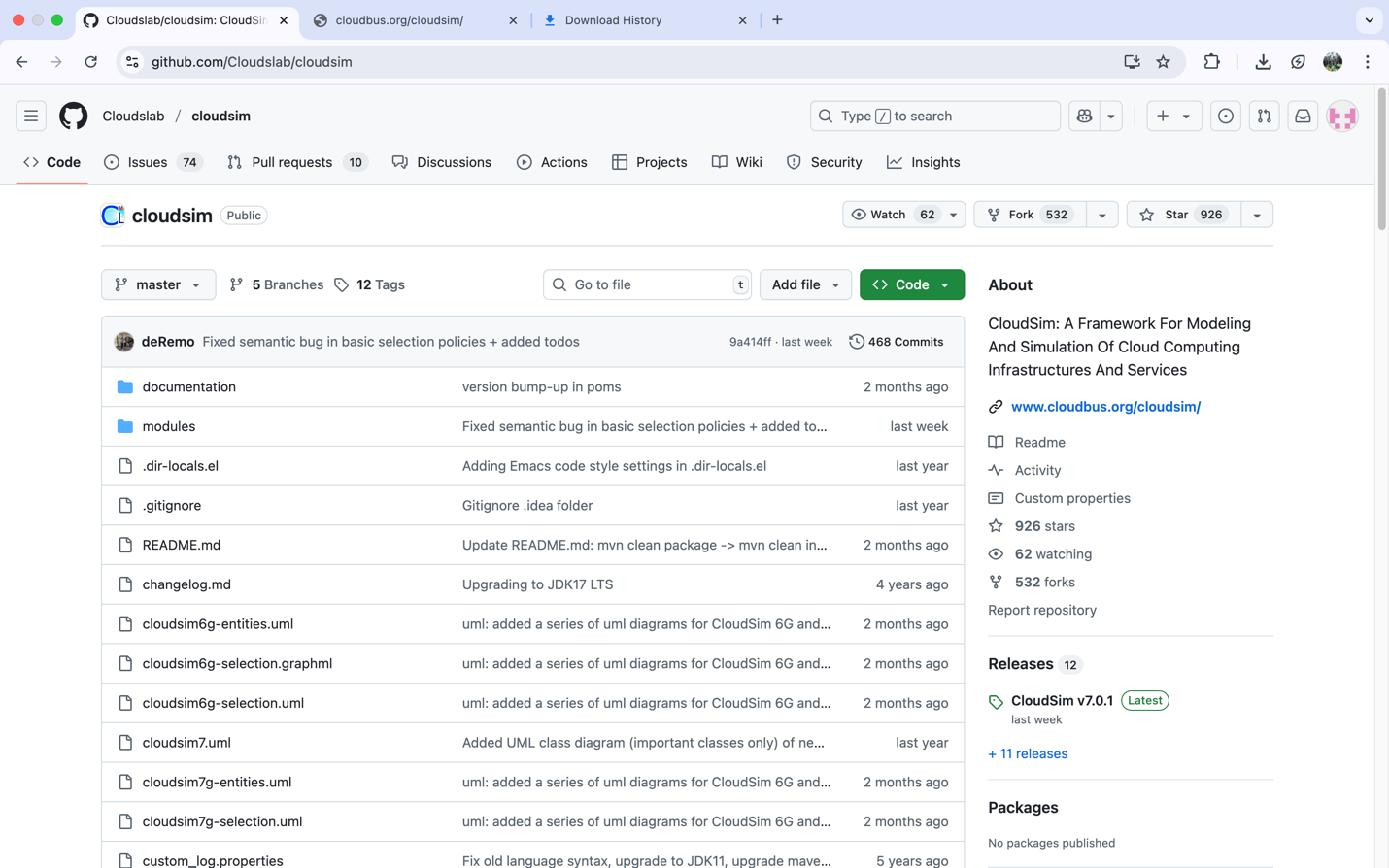
# 2. Set JAVA\_HOME and Update PATH Environment Variables

To use Java from the command line, you need to set the JAVA\_HOME environment variable and update your system PATH.  
- Setting JAVA\_HOME:  
 1. Open the Start Menu and search for 'Environment Variables'.  
 2. Select 'Edit the system environment variables'.  
 3. In the System Properties window, click 'Environment Variables...'.  
 4. Under System variables, click New...  
 5. Enter:  
 • Variable name: JAVA\_HOME  
 • Variable value: The path to your JDK installation (e.g., C:\Program Files\Java\jdk-21)  
 6. Click OK.  
- Updating PATH:  
 1. In the Environment Variables window, under System variables, find and select the Path variable.  
 2. Click Edit...  
 3. Click New and add: %JAVA\_HOME%\bin  
 4. Click OK on all windows to apply changes.  
- Verify Java installation:  
 Open a new Command Prompt and run: java -version

# 3. Install Apache Maven

Maven is used to build and manage CloudSim.  
- Download Maven:  
 Official site: https://maven.apache.org/download.cgi  
- Installation steps:  
 1. Download the Binary zip archive (e.g., apache-maven-3.9.4-bin.zip).  
 2. Extract the zip file to a directory, e.g., C:\Program Files\Apache\Maven.  
- Set MAVEN\_HOME and update PATH:  
 1. Open Environment Variables as before.  
 2. Under System variables, click New...  
 • Variable name: MAVEN\_HOME  
 • Variable value: path to your Maven folder (e.g., C:\Program Files\Apache\Maven\apache-maven-3.9.4)  
 3. Edit the Path variable and add %MAVEN\_HOME%\bin.  
 4. Click OK to save.  
- Verify Maven installation:  
 mvn -version

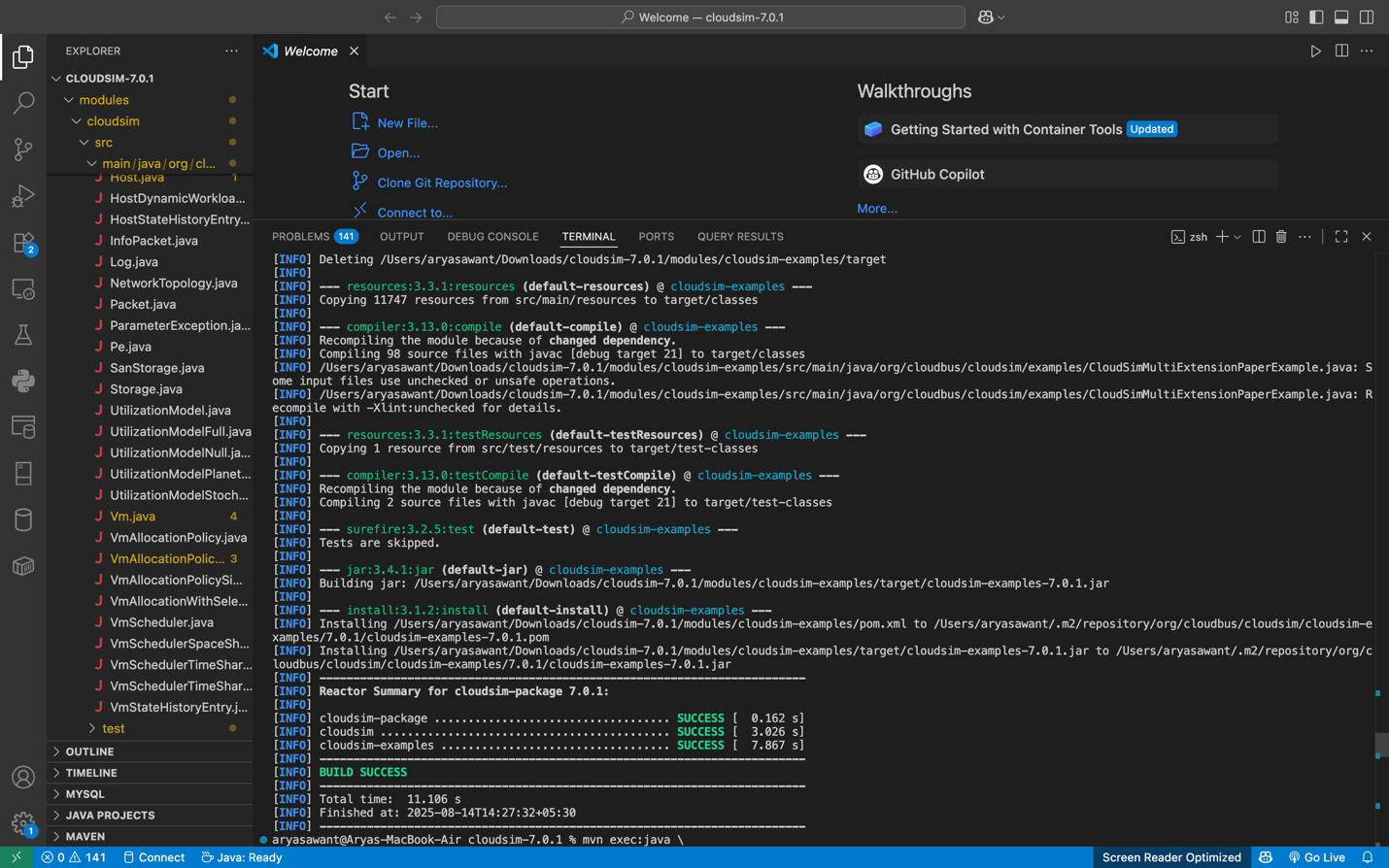
# 4. Download CloudSim Source Code

You can obtain CloudSim from GitHub in two ways:  
- Using Git (recommended if Git is installed):  
 1. Open Command Prompt.  
 2. Navigate to the directory where you want CloudSim.  
 3. Run: git clone https://github.com/Cloudslab/CloudSim.git  
- Using ZIP Download:  
 1. Visit https://github.com/Cloudslab/CloudSim  
 2. Click 'Code' → 'Download ZIP'.  
 3. Extract the ZIP file to a folder.

# 5. Open Command Prompt in CloudSim Directory

- Navigate to the root folder of the CloudSim project.  
- Open Command Prompt here or use 'Open in Terminal' in Windows 11.

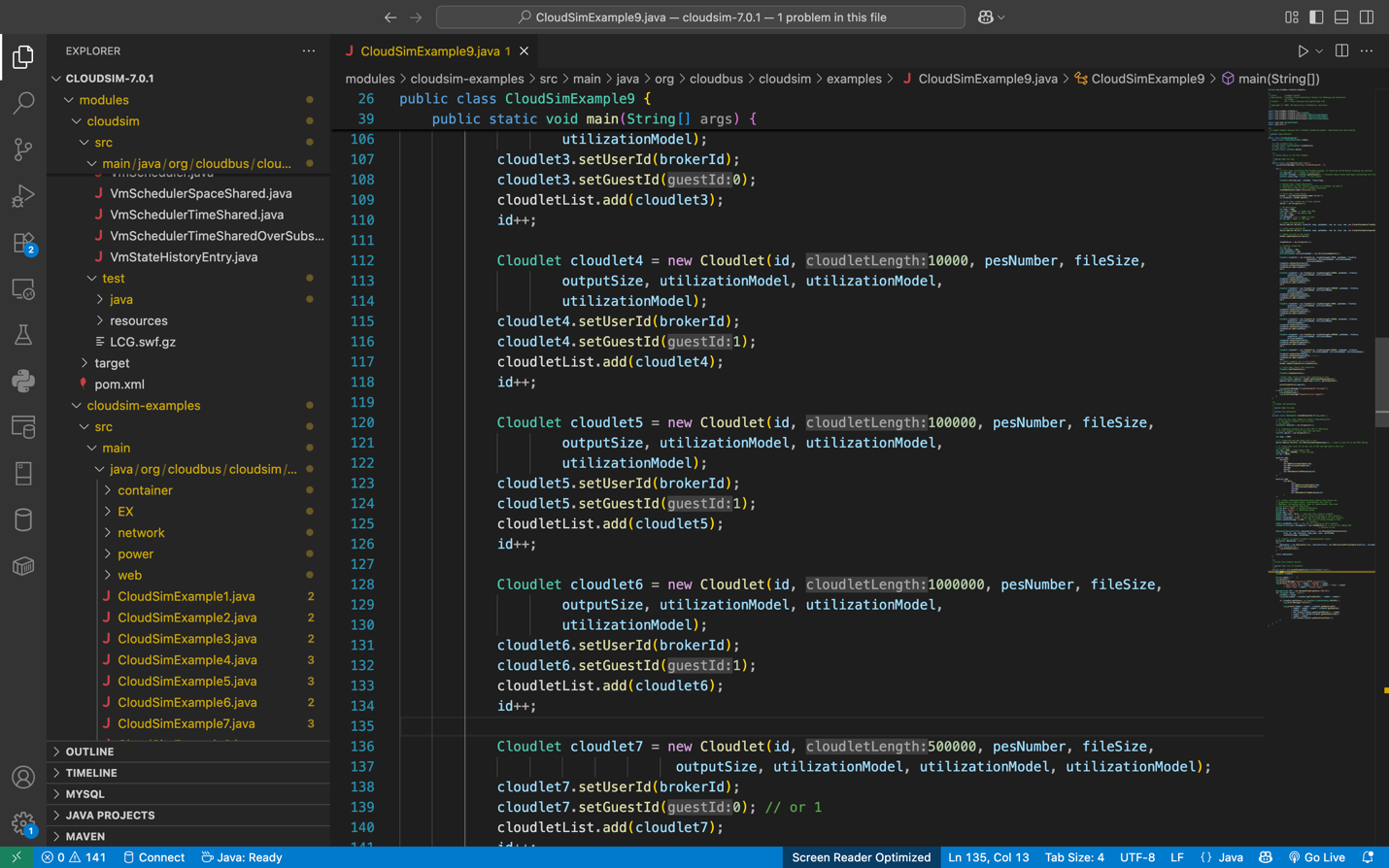
# 6. Build CloudSim Project with Maven

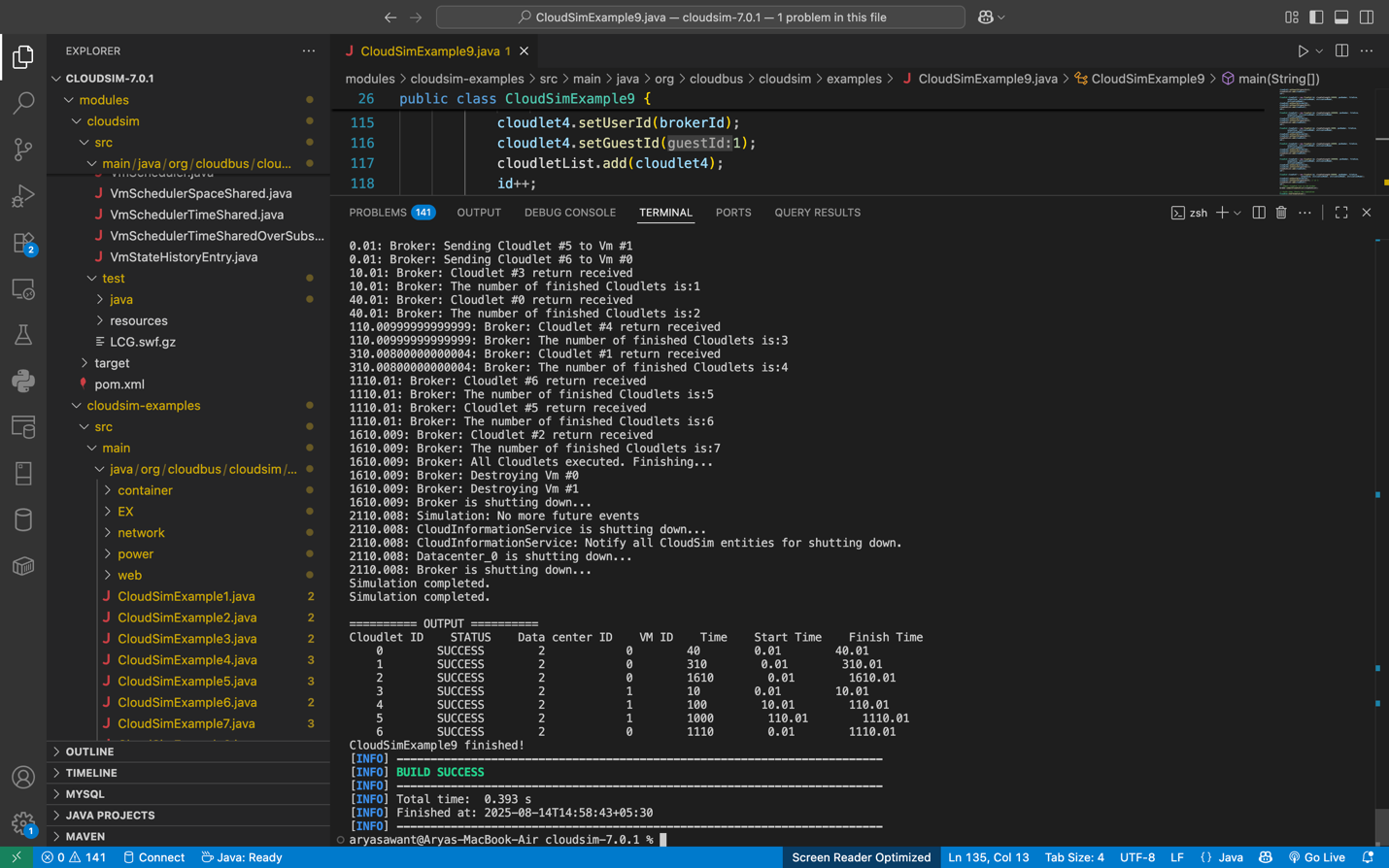
Run: mvn clean install  
- This downloads dependencies and compiles CloudSim.  
- If successful, Maven will display a BUILD SUCCESS message.

# 7. Run CloudSim Example (CloudSimExample9)

**To run CloudSimExample9:**mvn exec:java -pl modules/cloudsim-examples -Dexec.mainClass=org.cloudbus.cloudsim.examples.CloudSimExample9  
- This runs the simulation and prints results.

**To make changes in example 9:**

****

****

# Workflow Simulation using WorkflowSim (Basic Example 1)

## Objective

To simulate the execution of a scientific workflow (Montage\_100) using the **WorkflowSim toolkit** and observe how changing simulation parameters (like scheduling algorithm, number of VMs, etc.) impacts execution results.

## Simulation Setup

1. **Workflow Description**
   * Workflow is described in a **DAX file** (Montage\_100.xml).
   * Contains a set of interdependent tasks forming a Directed Acyclic Graph (DAG).
2. **Datacenter**
   * 1 Datacenter (Datacenter\_0) with **20 hosts**.
   * Each host has:
     + 2 processing elements (PEs) at 2000 MIPS each.
     + 2048 MB RAM.
     + 1,000,000 MB storage.
     + 10,000 Mbps bandwidth.
   * Cost model includes CPU, memory, storage, and bandwidth charges.
3. **VMs**
   * **20 CondorVMs** created.
   * Each VM:
     + 1 CPU (1000 MIPS).
     + 512 MB RAM.
     + 10000 MB image size.
     + Uses **Space-Shared scheduler**.
4. **Workflow Engine**
   * WorkflowPlanner → WorkflowEngine → Scheduler.
   * WorkflowEngine binds to datacenter and schedules tasks on VMs.
5. **Scheduling Algorithm**
   * Default: **MINMIN** scheduling.
   * Planning Algorithm: **INVALID** (so planner doesn’t override scheduler).
6. **Overheads & Clustering**
   * No overheads considered.
   * No clustering applied.

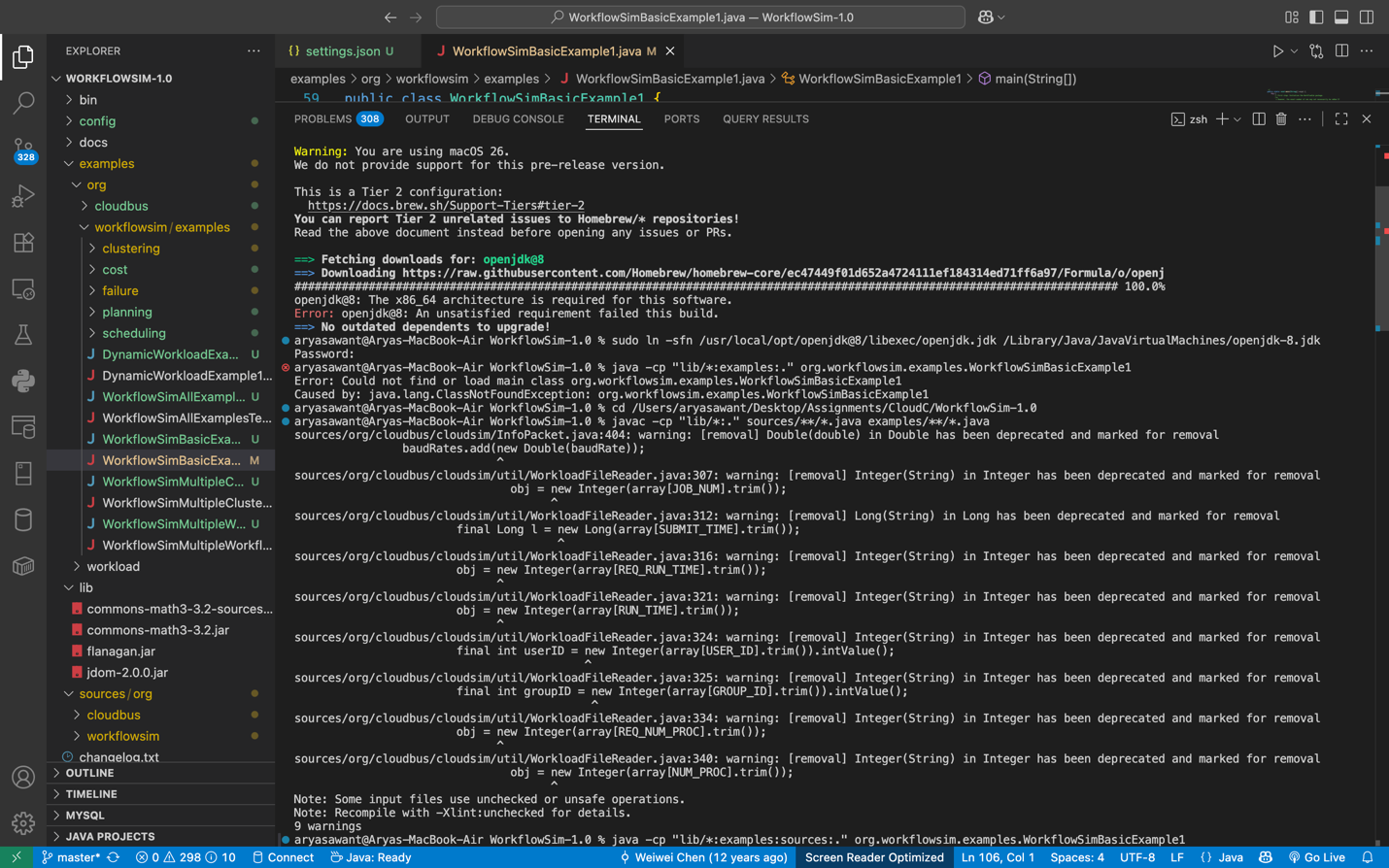
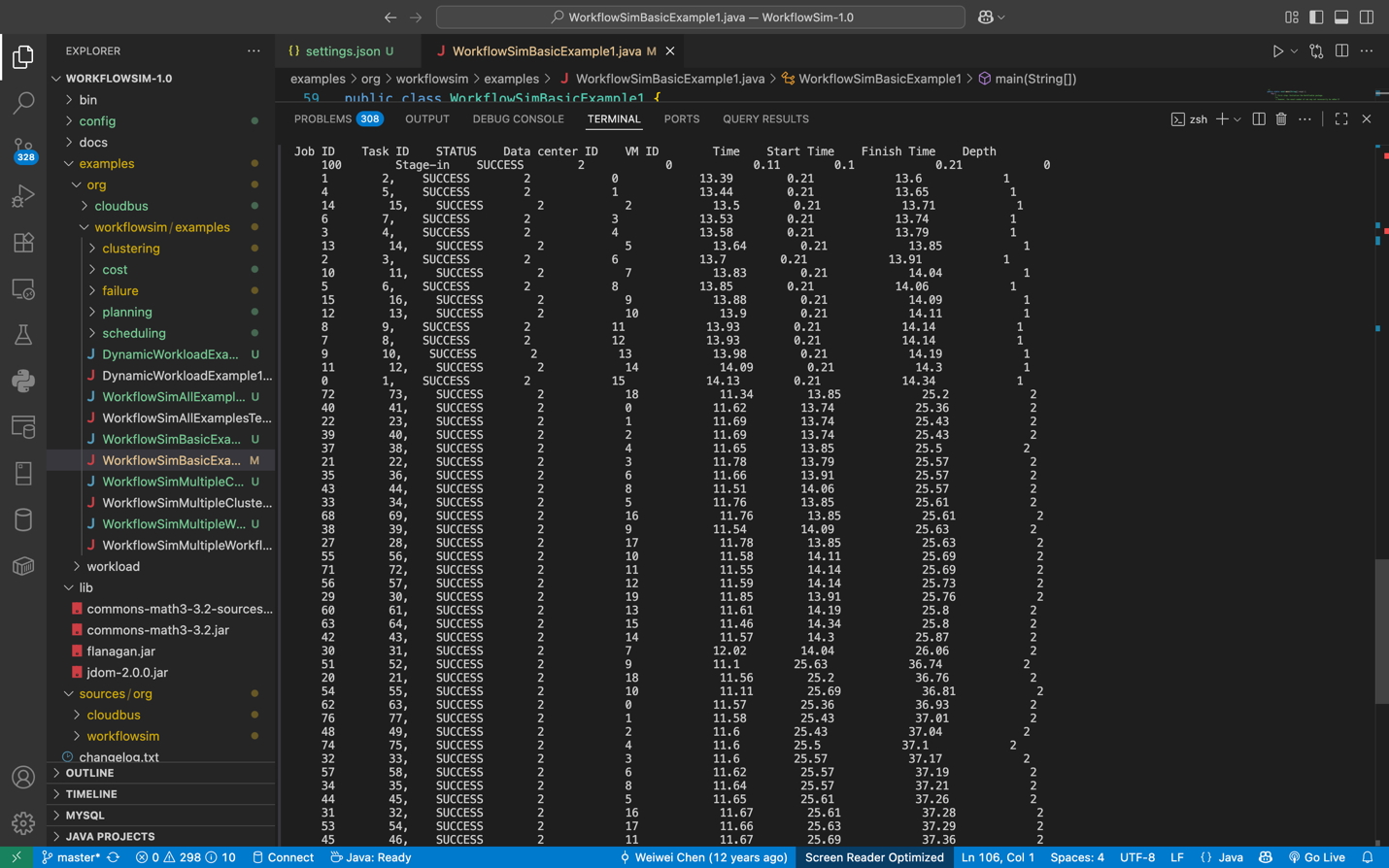
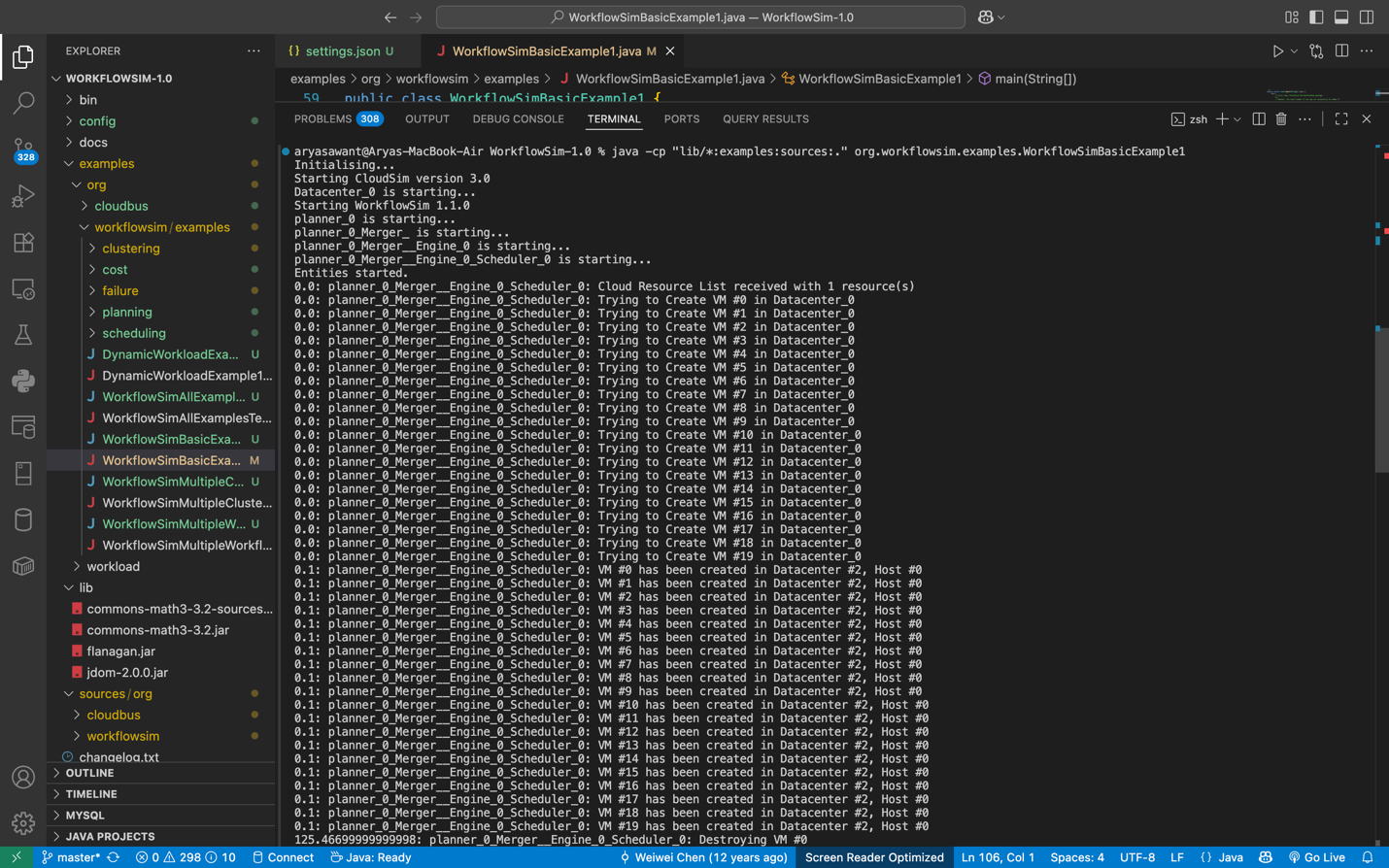
## Execution Workflow

1. Workflow DAX file is parsed into tasks.
2. Scheduler assigns tasks to available VMs based on scheduling policy.
3. Tasks are executed respecting dependencies.
4. Results are collected and printed (Job ID, Task ID, Status, VM ID, Execution Time, etc.).

## Modifications and Impact

### 1: Reduce Number of VMs

* Change: int vmNum = 10; // instead of 20
* Effect:
  + Fewer VMs → More tasks per VM → Increased waiting time.
  + Makespan increases.
  + Output shows tasks concentrated on fewer VM IDs.



OUTPUT:

After changes

