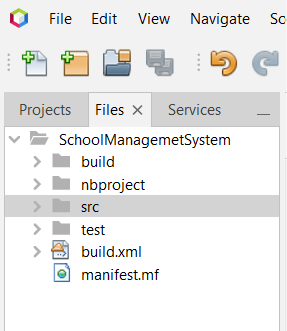
Building the project with Apache Ant

1. **Set up our project directory structure:**

Before we start building our project, we should set up a directory structure to organize our code and other files. A common directory structure for Java projects is to have a **src** directory for source code, a lib directory for third-party libraries, and a build directory for compiled code.

1. **Create a build.xml file:**

Apache Ant uses an XML file called build.xml to define the build process for our project. This file specifies the build targets, dependencies, and actions that Ant should execute when we run the build command.

1. **Define our build targets:**

A target is a specific task that Ant can execute. We should define targets for compiling our source code, running tests, and creating a deployable package of our project. We can also define targets for cleaning up old files, generating documentation, or any other tasks that are part of our build process.

1. **Specify dependencies:**

Our project may depend on external libraries or other resources that need to be included in the build process. We can specify these dependencies in our build.xml file using Ant's built-in dependency management system.

1. **Write our source code:**

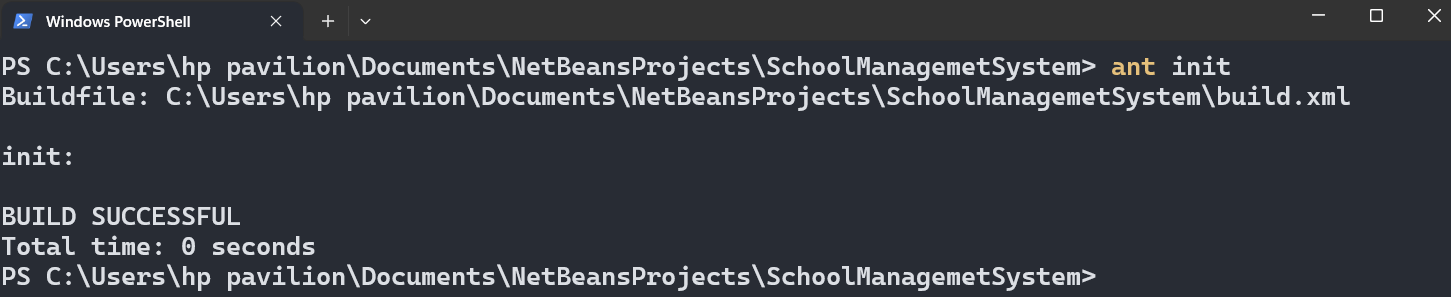
Once we have set up our project directory structure and build.xml file, we can start writing our source code. Be sure to organize our code into packages and classes that make sense for a school management system.

1. **Run the build:**

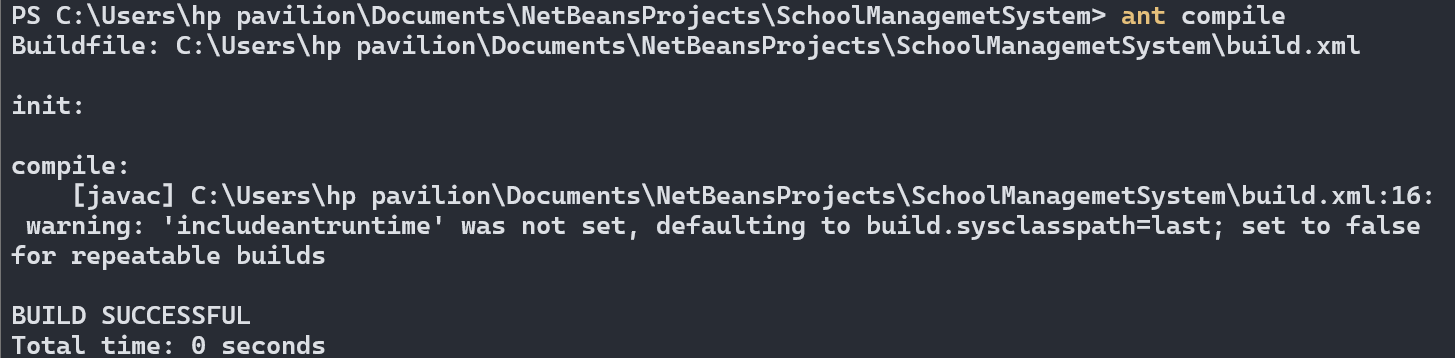
To build our project using Apache Ant, we simply need to run the ant command from the root directory of our project. This will execute the build targets that we have defined in our build.xml file and create a deployable package of our project.

Command: **ant [target name]**

* 1. Rung init target

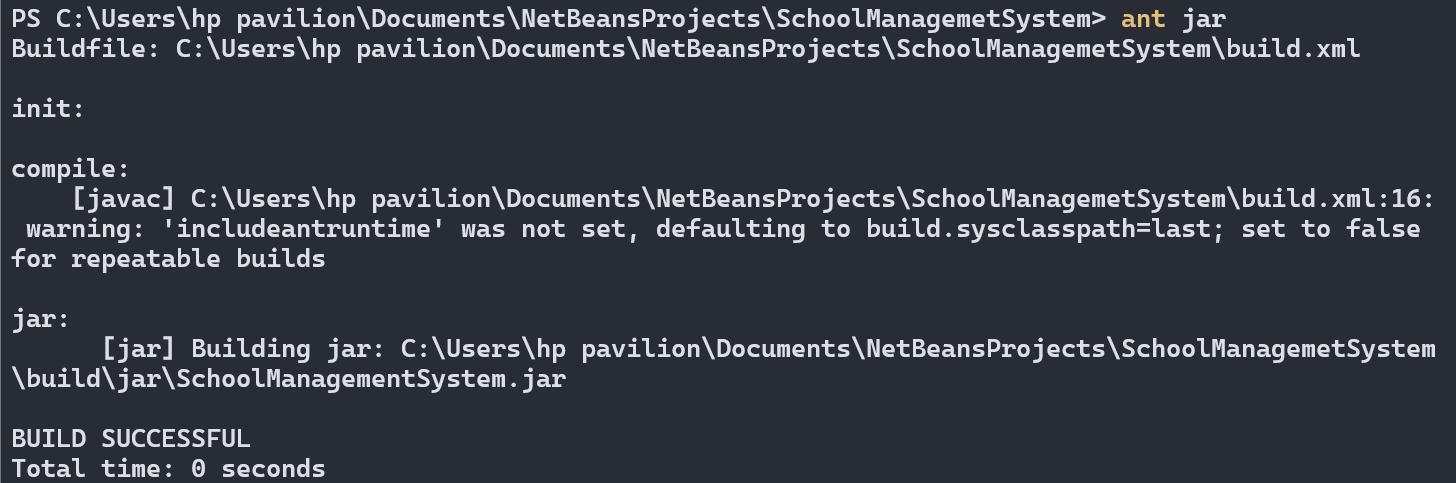
command: **ant init**

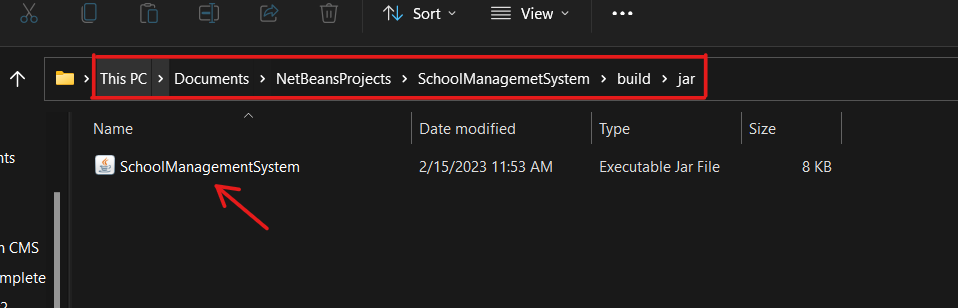
* 1. Run compile target

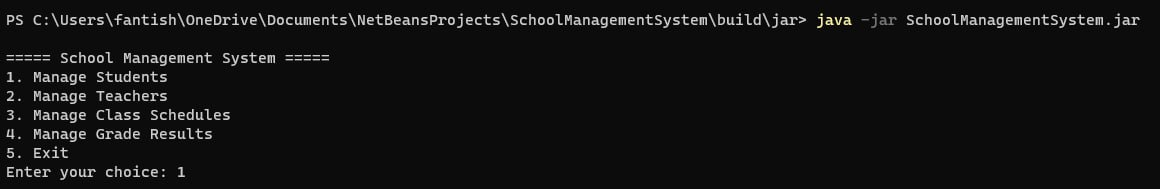
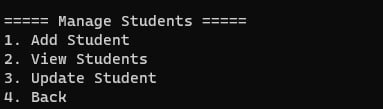
command: **ant compile**

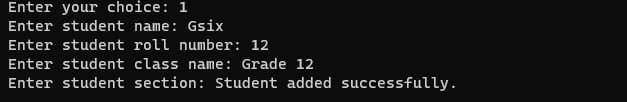
* 1. Run jar target

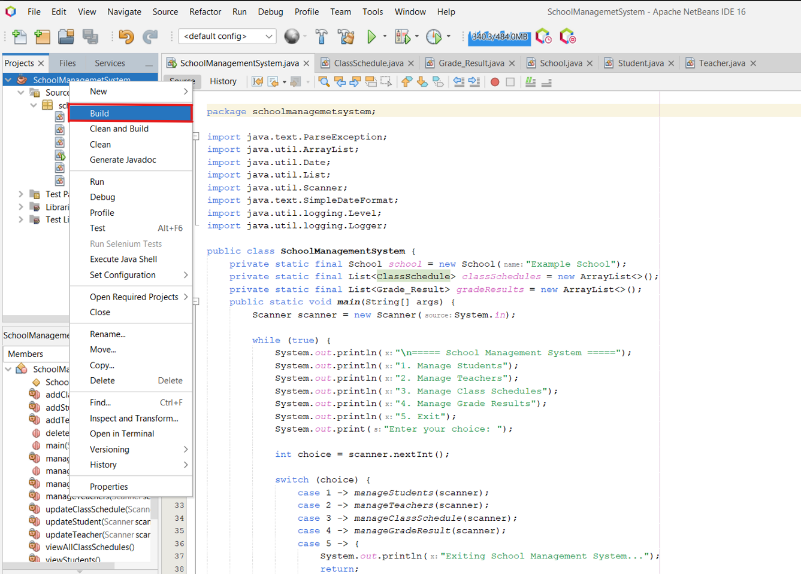
command: **ant compile**

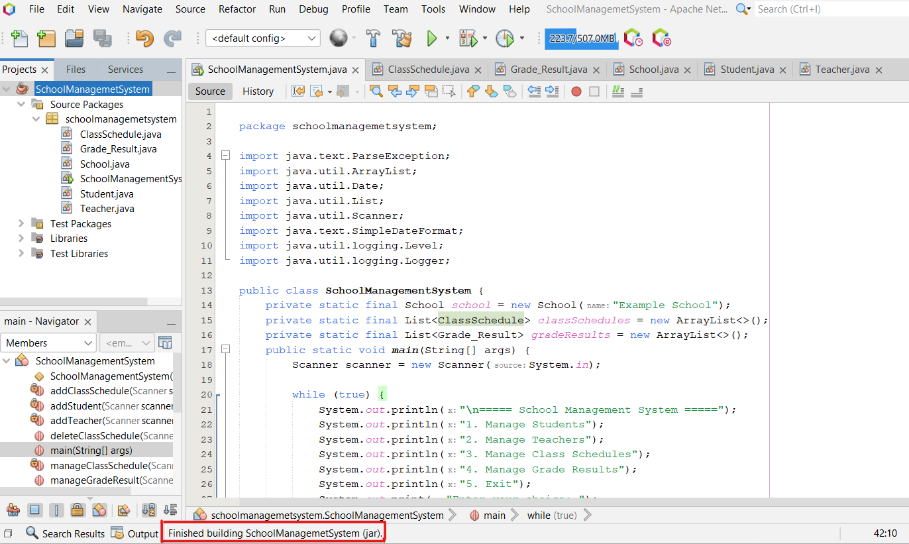


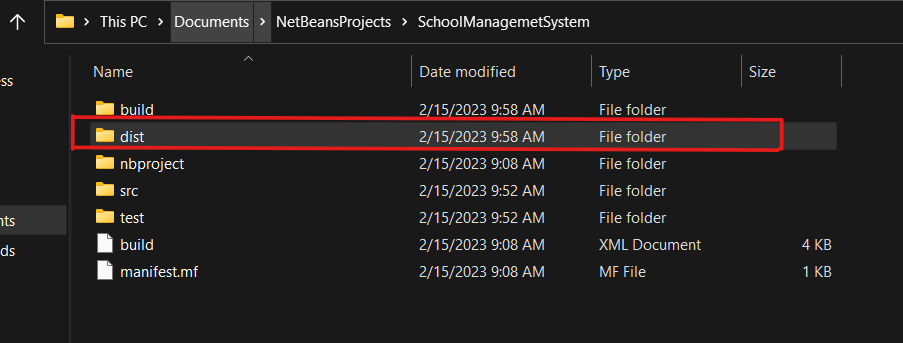
Then, check the presence of jar file, go to the root directory

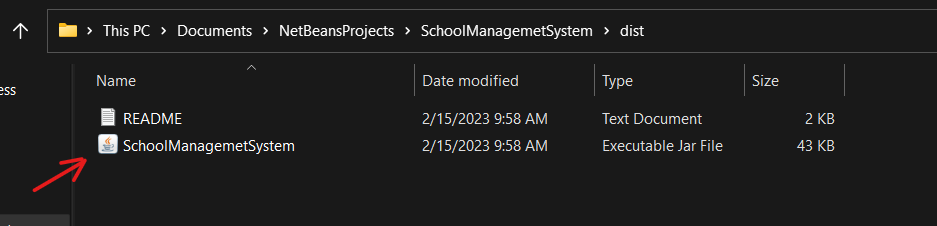
Then, let’s run the jar file:-

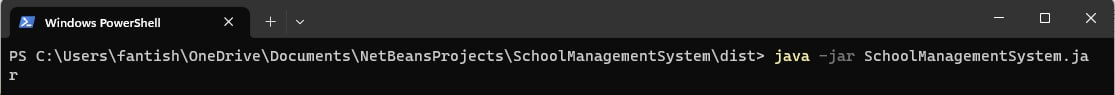
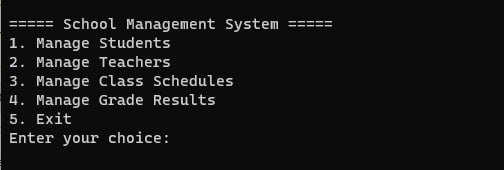






then, go to file explorer and find the project folder:



After we have checked the presence of the jar file go to command prompt and write the following command: