

Project

In this project you will implement the backend of a web application.

1. Submission instructions

The project should be submitted by pairs.

Deadline: TBD..

The project will be checked frontally, and the date of the frontal checking will be published soon.

You should submit zip file with your project to the model.

The name of your project should be of form id1_id2 where id is your ID number.

2. Development instructions

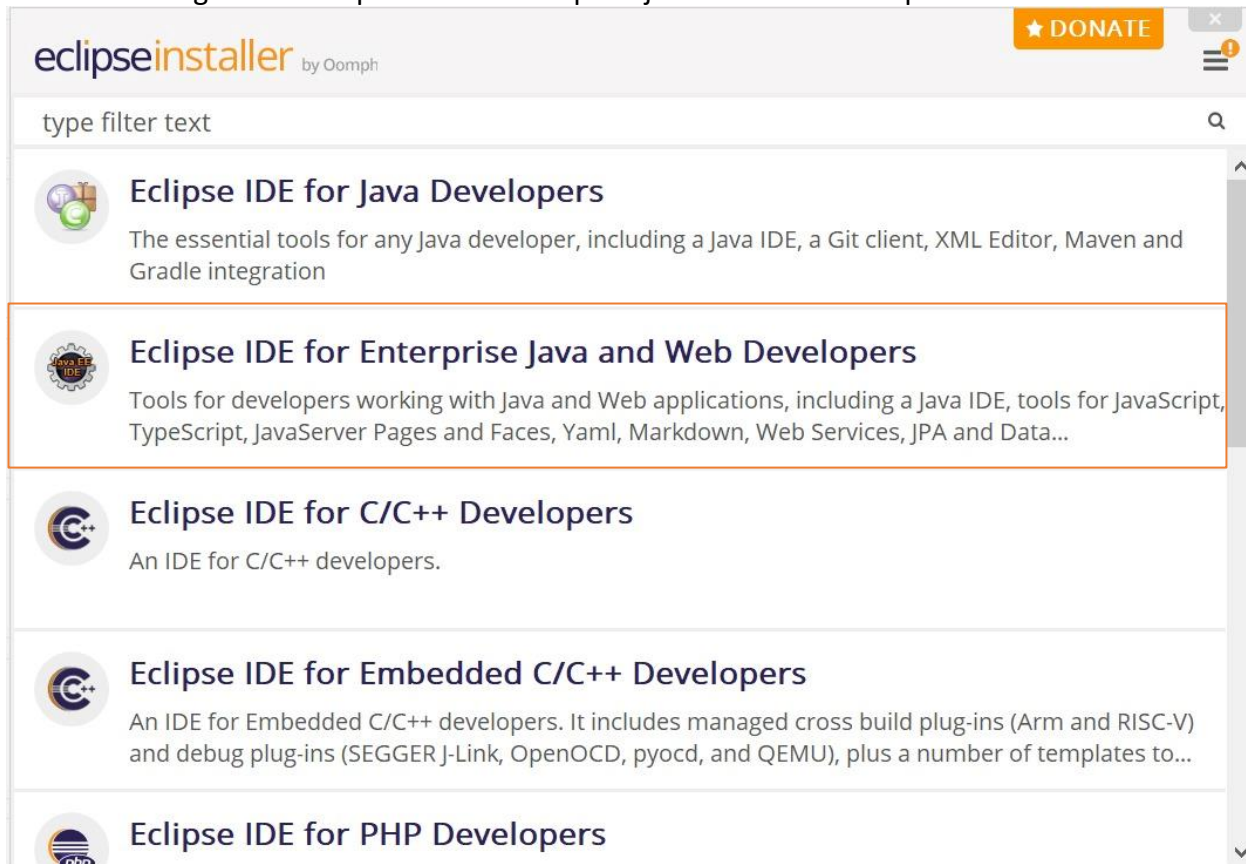
Make sure you have the table from the previous assignment (MediaItems).

2.1. Preparations

2.1.1. Download Eclipse:

<https://www.eclipse.org/downloads/download.php?file=/oomph/epp/2023-03/R/eclipse-inst-jre-win64.exe>

When installing select "Eclipse IDE for Enterprise jav and Web Developers"



2.1.2. Download Apache Tomcat:

<http://tomcat.apache.org/>

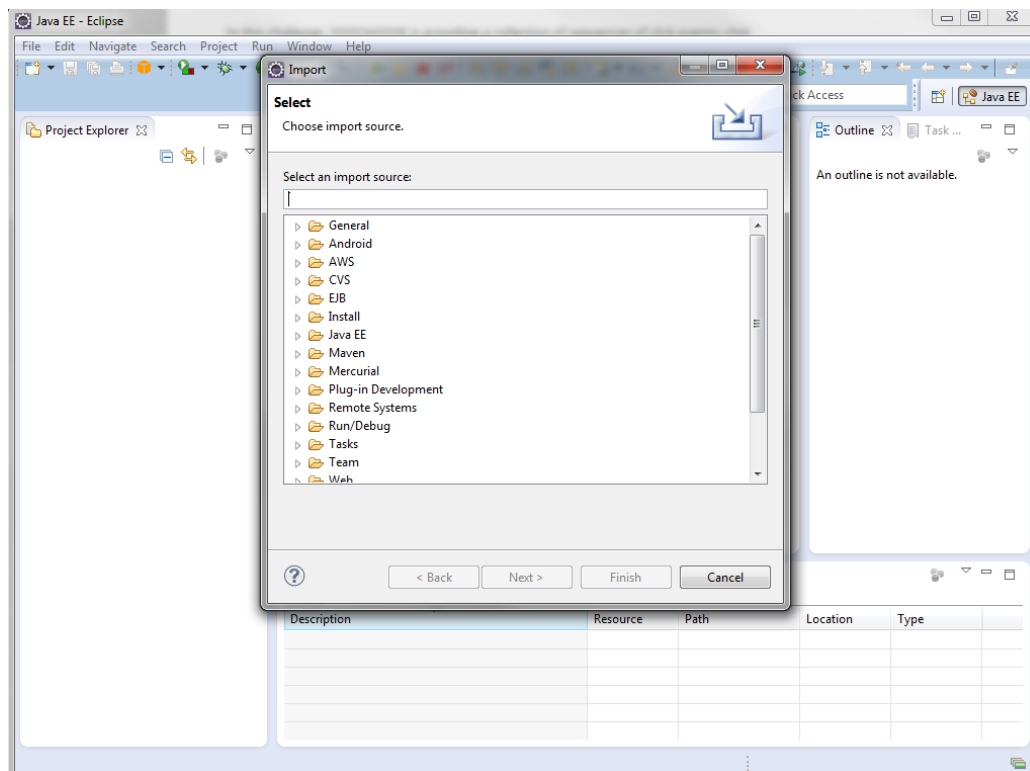
Choose core version that fits your OS, Eclipse and java version,

No need to install only unpack.

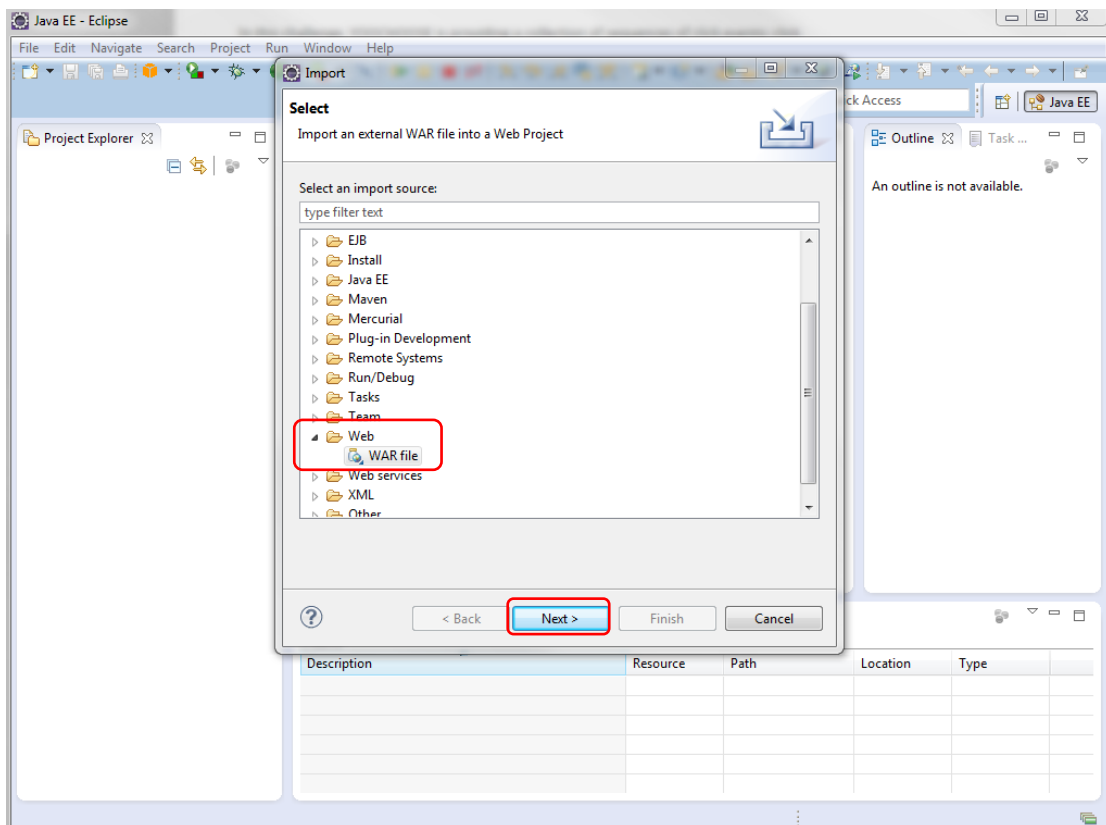
2.1.3. Download project war file from Moodle.

2.1.4. Import project to eclipse

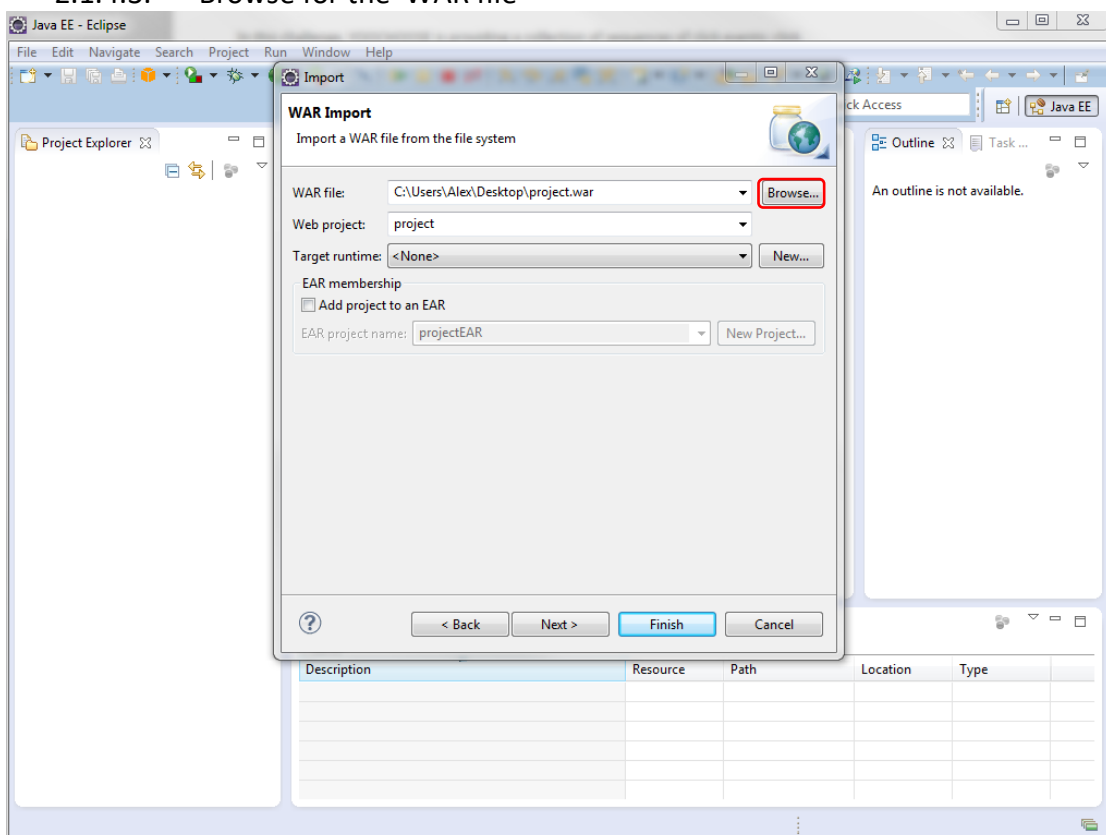
2.1.4.1. Open Eclipse, click on "File" -> "Import..."



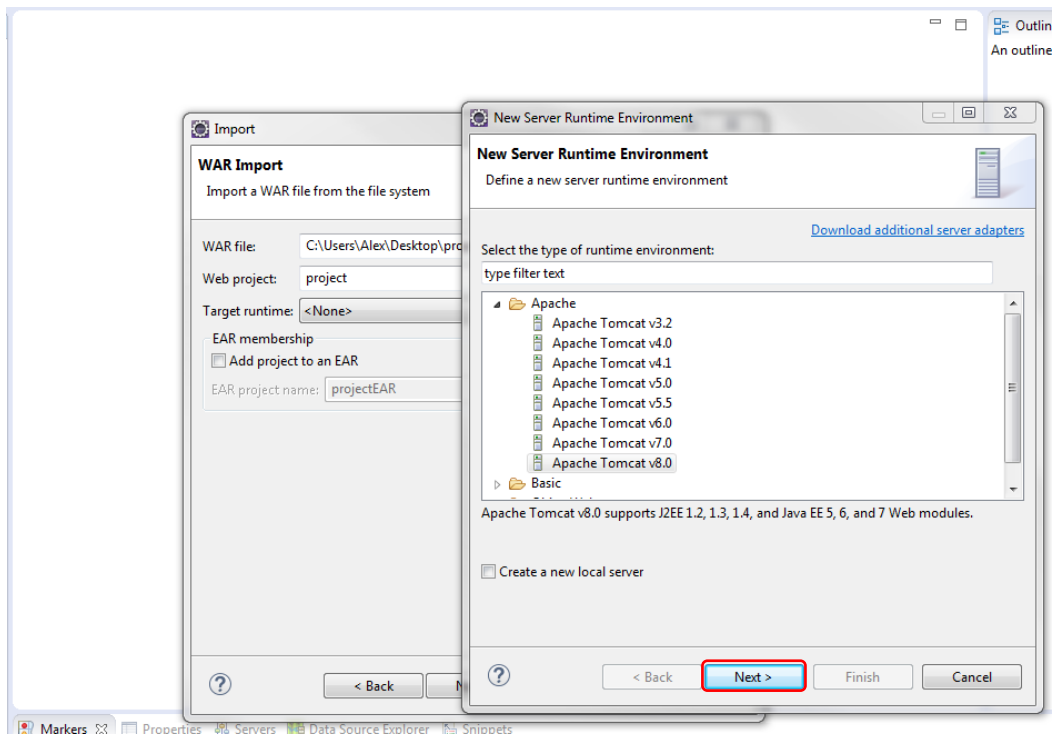
2.1.4.2. Choose "Web" -> "WAR file" -> "Next"



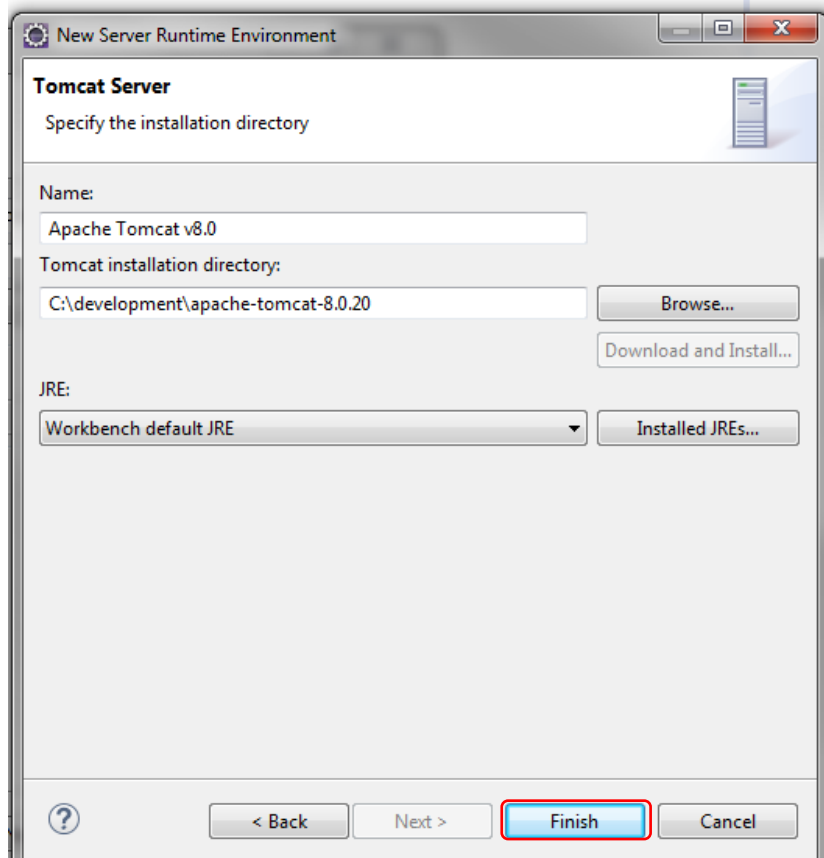
2.1.4.3. Browse for the WAR file



2.1.4.4. On the line of Target runtime click on "New..." and choose appropriate Apache Tomcat server, click on "Next"



2.1.4.5. Browse for Tomcat installation directory and click on "Finish"



2.1.4.6. Don't choose nothing and again click on "Finish"

2.2. Java

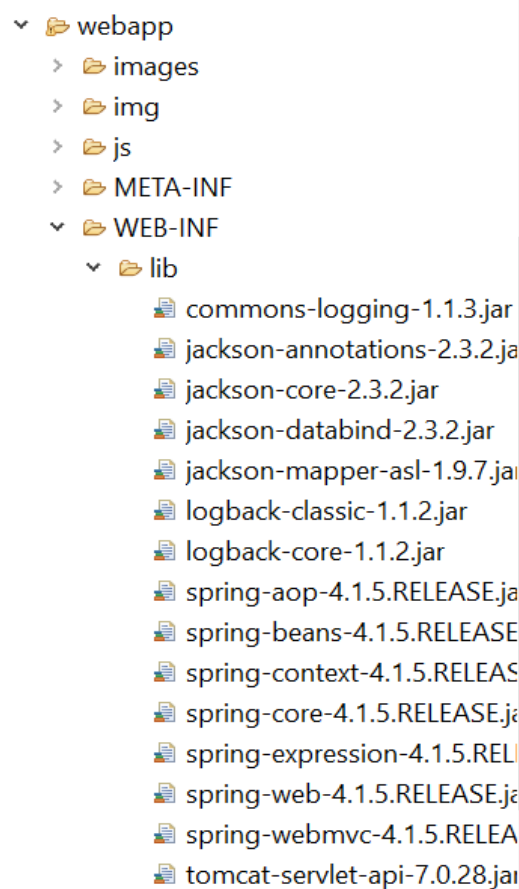
- Implement all functions in *RegistrationController*, *ItemsController* and *HistoryController* use **Mongo DB**. Please read **carefully** the instructions written in Java Doc comments in these classes.

Example of url to fillMediaItemsFromUrl function (in itemsController):

https://drive.google.com/uc?export=download&id=1nu33_hB7kB38g75XeCQMqnzpBqFb4t40

- If you need to use any external libraries (jars), for example – for the MS SQL server & MongoDB, you should copy them physically to the folder "webapp/WEB-INF/lib"

Don't touch the jars that already exists over there



- Working with Mongo:
Import –
`import com.mongodb.*;`
Useable object types:
MongoClient – set the connection to MongoDB
DB – mongo DB object
DBCollection – mongo collection
DBCursor – cursor object
BasicDBObject – like PreparedStatement in JDBC

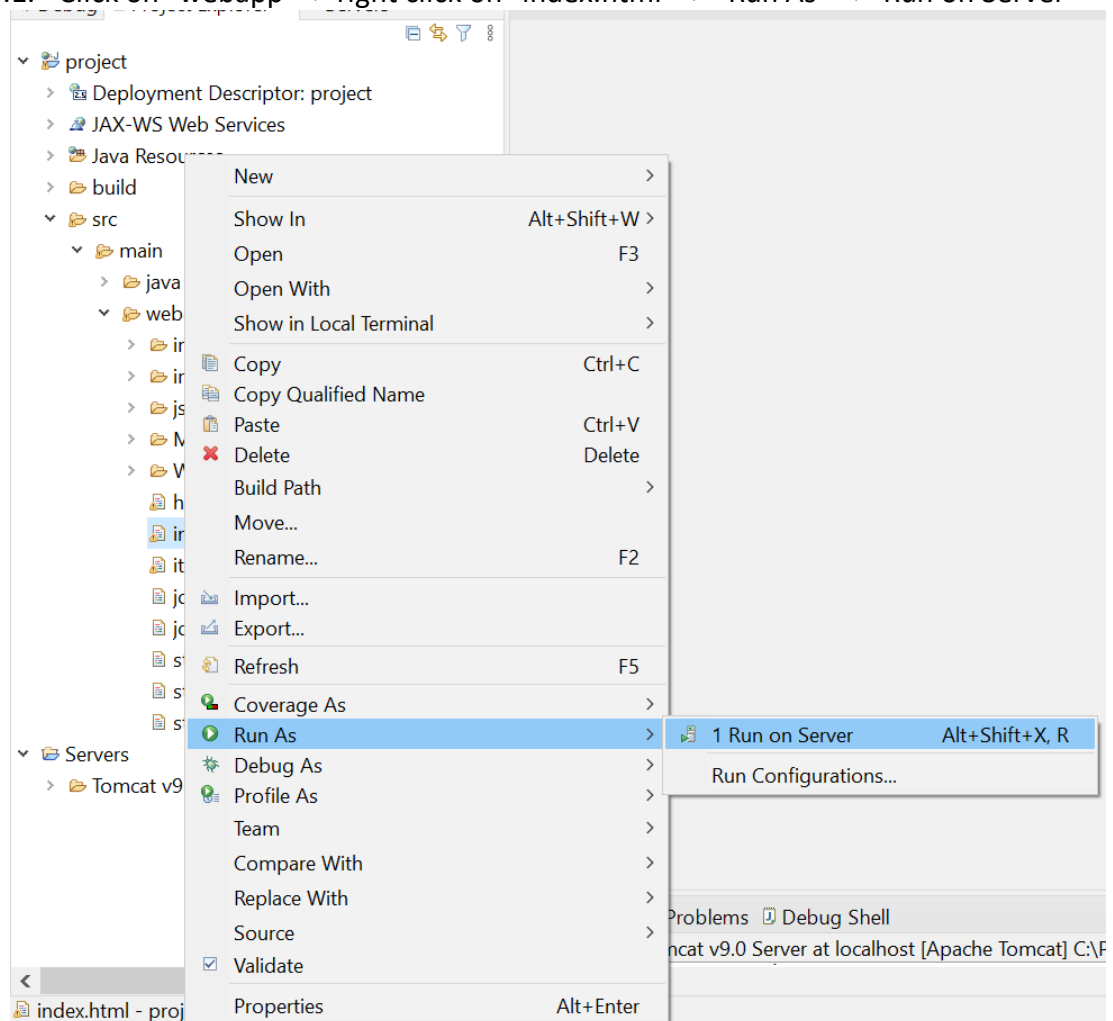
Connect to MongoDB using Java:

After adding the jar file, use the following code to set connection to Mongo, query "Users" collection to find all the users with specific username and password, and get results as cursor:

```
MongoClient mongoClient = new MongoClient("localhost", 27017);
DB db = mongoClient.getDB("BigDataProject");
DBCollection users = db.getCollection("Users"); // get Users collection
BasicDBObject myQuery = new BasicDBObject();
myQuery.put("username", username);
myQuery.put("password", password);
DBCursor cursor = users.find(myQuery); // Send query to Mongo, get the results
```

2.3. Running Web application in Eclipse

2.3.1. Click on "webapp" -> right click on "index.html" -> "Run As" -> "Run on Server"



2.3.2. Choose your tomcat version and click on "Finish"

Run On Server

Run On Server

Select which server to use

How do you want to select the server?

☒ Choose an existing server

☐ Manually define a new server

Select the server that you want to use:

type filter text

Server	State
▼ 📁 localhost	
📄 Tomcat v9.0 Server at localhost	🛑 Stopped

Apache Tomcat v9.0 supports J2EE 1.2, 1.3, 1.4, and Java EE 5, 6, 7, and 8 Web modules.

Columns...

☐ Always use this server when running this project

?

< Back

Next >

Finish

Cancel

Good Luck !

Noy Cohen-Shapira and Alon Malach