Front-end:

- Install nvm (node version manager), before this step make sure you have completely uninstalled node.js from your machine. Nvm will allow you to manage multiple node versions at once; go here and download (https://github.com/coreybutler/nvm-windows/releases), choose 'nvm-setup.zip' extract and install. (by the end of this step nvm should have been added to your path)
- Close all previously opened command lines, and open a new command prompt with administrator privileges (necessary for all future installations):
    - From the command line test if nvm is installed by running: ***nvm -v***
    - Install node version 10.6.0: ***nvm install 10.6.0***
- Go to the front-application directory and place yourself on the command line at the same position (this can be achieved by navigating through the windows, once on the desired window write cmd in the navigation bar then enter, a cmd in the correct location will be opened; make sure the command line is opened with admin privileges
- In the content of the folder you should have a 'bower.json' file it contains all definitions for the front-end dependencies
- To use this file you need to install the 'bower' utility globally: ***npm install -g bower***
- After installation is finished bower should have been added to your path to check run ***bower -version***
- Install dependencies by running this command (this will take a while) : ***bower install***
- Once installation is done you can serve the application to see how it looks but it will not work (yet)
- To serve the application you need to install a local server: ***npm install http-server***
- Serve the application (with no caching) using the port 8050 the back-end will expect calls from this address: ***http-server . -p 8050 -c-1***
- If everything went well a message saying 'Starting up http-server...' will show on the command line

Back-end:

- Add java jdk to your path !
- Download Maven (https://maven.apache.org/download.cgi) and add it to your path (very important check web to see how, you can use RapidEE software to add variables to your path) : apache-maven-3.5.4-bin.zip
- Extract whereve you want but make sure the location will not move otherwise path variable will stop working ; add this folder to your path '\apache-maven-3.5.4-bin\apache-maven-3.5.4\bin' it contains a .bat file called mvn that we will be called from the command-line to install back-end dependencies
- Assuming you have the java jdk added to your path run ***mvn -version*** to check if maven was added to your path and can be called
- Go to root of the project you should see a pom.xml file, place yourself on the command line and run ***mvn install -DskipTests***
- This will install dependencies of this project

Data base configuration:

- Before launching the back-end application you need to configure the database connection in the application
- Open ‘db.properties’ file that you find in the resources package of the application
    - `db.driver=com.mysql.jdbc.Driver`
    - `db.url=jdbc:mysql://192.168.99.100:5432/University`
    - `db.user=root`
    - `db.pass=`
- Change the values to that of your database (url, user and password if any); in my database I was using the user ‘root’ with no password
- Run data.sql in your database before launching the application
- Run the application using your IDE (better for debugging) or using the commad line by going into the target folder and running : ***java -jar demo-0.0.1-SNAPSHOT.jar***

Your back-end should be up and running (on port 8080) and ready to receive calls from the front-end application (port 8050) (your db should not use these ports, typically my-sql db uses ports 5432)

You are ready to test the application
enjoy