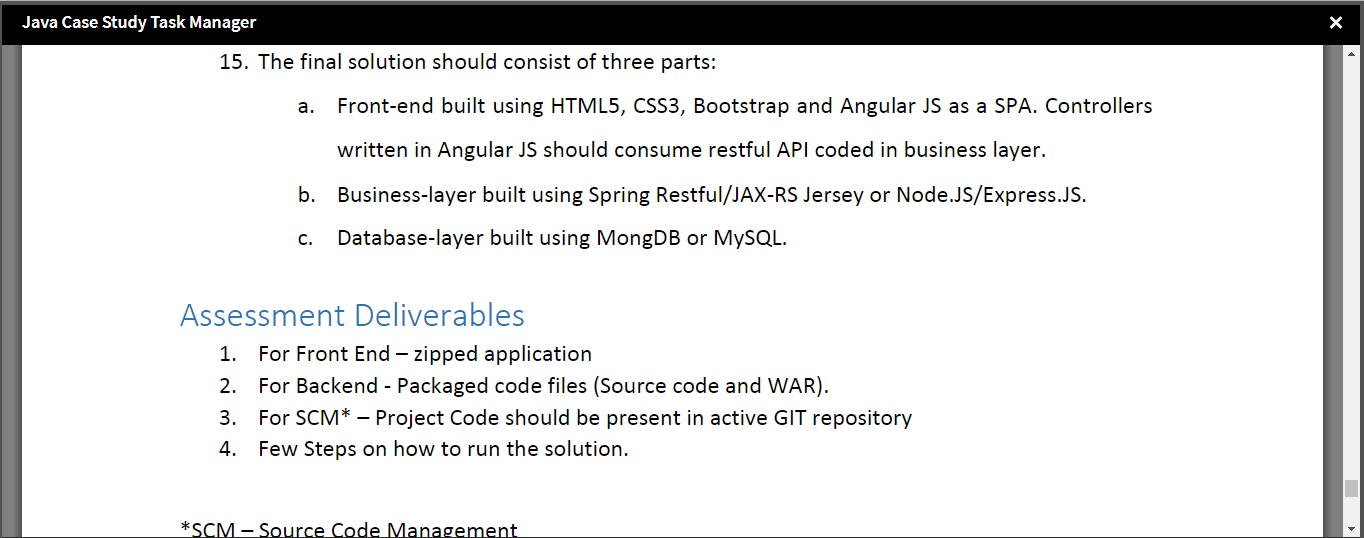
Requirements for the Capsule Program - 

**Core Content**

1. UI Layer Submission
2. Middleware Server side Services Submission
3. Database connectivity, Integration &Performance Testing Submission
4. Deployment Submission
5. **Final Submission - All the individual Step 1 to Step 4 are given in detail below which should cover the final submission. Also attachments are given in the end of this document for the source code.**

# All Information on #1 UI Layer Submission

# App Name - tasktracker-fsd-ui

If you are running the application from localhost/Docker(@latest version), the url needed to be hit to access the application is - [http://localhost:4200](http://localhost:4200/)

If you are running the application from Docker Tool Box for windows (for older windows versions), the url needed to be hit to access the application is - [http://192.168.99.100:4200](http://192.168.99.100:4200/)

## The following should be executed only when your service layer is up and running if data does not exist in the following order:

1. If you do not find any values in the database for Parent Task Dropdown then please hit the url - For localhost/Docker(@latest version) - [**http://localhost:8080/tasktrackerbackend/parent/dump**](http://localhost:8080/tasktrackerbackend/parent/dump) or For Docker Tool Box for windows (for older windows versions) - <http://192.168.99.100:8080/tasktrackerbackend/parent/dump>
2. If you do not find any values in the database for View Task Link then please hit the url - For localhost/Docker(@latest version) - [**http://localhost:8080/tasktrackerbackend/task/dump**](http://localhost:8080/tasktrackerbackend/task/dump) or For Docker Tool Box for windows (for older windows versions) - <http://192.168.99.100:8080/tasktrackerbackend/task/dump>

Information on docker commands and execution -

# Docker image is built and pushed to docker repository. Just incase the dist folder is not there, go to UI project root folder and execute the build command: ng build

# Then perform the following docker commands to execute in docker.

## Docker Stuff for UI Layer - Task Tracker Project

# General Commands to be executed in Docker related to push/build/run-

docker build -t your\_image\_name your\_repo\_url

docker tag your\_image\_name docker\_username/your\_image\_name

docker push your\_image\_name docker\_username/your\_image\_name

docker run -d -p 80:80/tcp -p 80:80/udp --name your\_image\_name your\_image\_id\_value

# Example for push/build/run-

docker build -t fsd-img-ui https://github.com/fsduseriiht/tasktracker-fsd-ui.git

docker tag fsd-img-ui amitabhadockerwork/ui-layer-fsd

docker push amitabhadockerwork/ui-layer-fsd

docker run -d -p 80:80/tcp -p 80:80/udp --name fsd-img-ui\_running 127186b66ec9

# All Information on #2 Middleware Server side Services Submission

# App Name - tasktracker-fsd-service

## Information on the services exposed -

#### API Method Reference

{{baseurl}} = the context url with dns and port and application context

Example -

Local PC/Docker - http://localhost:8080/tasktrackerbackend/

Docker Tool Box for older windows - http://192.168.99.100:8080/tasktrackerbackend/

#### Task Entity-methods

GET METHOD - **{{baseurl}}task/dump** - Used to create the TASK\_TABLE database dump with sample values when/if H2 db is used (only if table does not exist - also first execute dump of PARENT\_TASK\_TABLE if not exist)

GET METHOD - **{{baseurl}}task/list** - is used to display all records in task table db

POST METHOD - **{{baseurl}}task/** - is used to create a single task record in task table in db

PUT METHOD - **{{baseurl}}task/:taskId** - is used to update the task table single record in db

DEL METHOD - **{{baseurl}}task/:taskId** - is used to delete record from TaskTable in db

#### Task Parent Entity - Methods

GET METHOD - **{{baseurl}}parent/dump** - Used to create the PARENT\_TASK\_TABLE database dump with sample values when/if H2 db is used (only if table does not exist)

GET METHOD - **{{baseurl}}parent/list** - is used to display all the ParentTask values in the db

POST METHOD - **{{baseurl}}parent/** - is used to create a parent task in the db

PUT METHOD - **{{baseurl}}parent/:parentId** - is used to update a parent task in the db

DEL METHOD - **{{baseurl}}parent/:parentId** - is used to delete a parent task from the db

### useful links related to docker:

<https://docs.docker.com/docker-cloud/builds/push-images/>   
<https://docs.docker.com/docker-hub/builds/>   
<https://hub.docker.com/r/amitabhadockerwork/>

### Docker commands

docker pull amitabhadockerwork/ui-layer-fsd   
docker pull amitabhadockerwork/service-layer-fsd

## Docker for Service Layer - Task Tracker Project

# General Commands to be executed in Docker related to push/build/run-

docker build -t your\_image\_name your\_repo\_url

docker tag your\_image\_name docker\_username/your\_image\_name

docker push your\_image\_name docker\_username/your\_image\_name

docker run --rm -i -t -d -p 8080:8080/tcp -p 8080:8080/udp --name your\_image\_name your\_image\_id\_value

# Example for push/build/run-

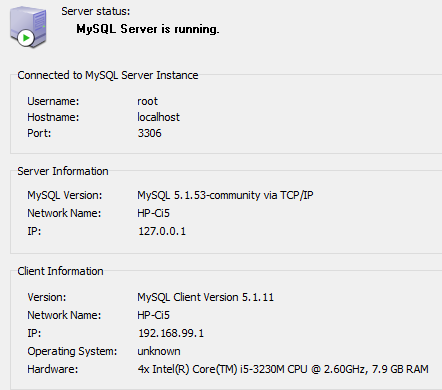
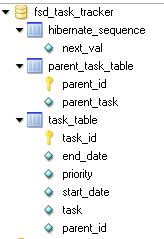
docker build -t service-layer-fsd https://github.com/fsduseriiht/tasktracker-fsd-service.git

docker tag service-layer-fsd amitabhadockerwork/service-layer-fsd

docker push amitabhadockerwork/service-layer-fsd

docker run --rm -i -t -d -p 8080:8080/tcp -p 8080:8080/udp --name service-layer-fsd\_running 6a35e567faf4

# All Information on #3 Database connectivity, Integration &Performance Testing Submission

**Note** - For Database, if you do not have MySql Installed in your system then you can use the H2 Database server profile also. Only you have to hit the the following urls if you do not find any data at application start (in the order given below)-

1. http://localhost:8080/tasktrackerbackend/parent/dump
2. http://localhost:8080/tasktrackerbackend/task/dump

# All Information on #3 Deployment Submission

For Deployment you can use Docker as mentioned above.

You can also deploy locally as per following steps.

**For service layer -**

1. Pull from the git repo - **https://github.com/fsduseriiht/tasktracker-fsd-service.git**
2. Build using maven (mvnw) which is given as a part of the source code package. Set java home folder only if necessary in the mvnw.cmd file as per your own system. [**mvnw clean install**]
3. Go to target folder generated as a part of build result of the previous step and then execute the war file generated as a result of the build process using [**java -jar <war file name>].**
4. Eg. - java -jar tasktracker-backend-0.0.1-SNAPSHOT.war
5. Check using post man or the logs in cmd prompt if the service is up or not.
6. Now start the ui application as follows.

**For UI - Layer -**

1. Pull from git repo - **https://github.com/fsduseriiht/tasktracker-fsd-ui.git**
2. The build folder is already available in the folder from repo. All you need is to run **ng serve** command to run the UI application from command prompt
3. Hit the url - **http://localhost:4200**

**Attachments -**

* UI Layer - 
* Service Layer - 
* DB Layer - Not needed as application will create the db/table if not exist.
* PostMan configuration script - 

Capsule Program Git URL - **https://github.com/fsduseriiht/capsule-program.git**