# Jenkins & Docker Assignment (MEAN)

1. Consider the attached project. It is the CRUD app created using node with support for mongodb database.
2. Open it is in Visual Studio Code, setup dependencies and configure it locally to run it as node App.
3. You may need to configure MongoDB credentials and database name to connect with local mongodb server.
4. Execute the project locally and access the app at <http://localhost:portnumber>
5. Once, it is working fine in local development environment; do the following as a part of deploying the app to docker as container:
   1. Push the app source in internal GIT server. Pl. ask your mentor for the Internal GIT server URL.
   2. Configure Jenkins locally to pull the source from internal GIT repository
   3. Jenkins should build the project and create the deployable
   4. Create a docker file and docker-compose.yaml such that, when you run the following command:
      1. docker-compose up
      2. It should deploy the attached project at <http://localhost>
   5. Hints:
      1. Docker Compose file should:
         1. Create ngix container
         2. Create mongodb container
         3. Configure attached “app” to run in docker containers
6. CRUD app in Node and MongoDB
   1. 
7. Once the app is deployed in docker; do the following to demonstrate the Continuous Deployment:
   1. Change the source code in index.jade such that it says “New Employee” in place of “Add New Employee”
8. Make it sure that project is running locally in development environment without errors.
9. If it running locally without errors, push the changes to the internal GIT repository which was connected in 5.a
10. If 5.b was done correctly, Jenkins will automatically pull the code updates from internal GIT repo and build and deploy the project with updated code.
11. Now, when you visit <http://localhost>; you should see the changes in the browser window.