Tasks:

Use Docker Compose to manage a multi-service application.

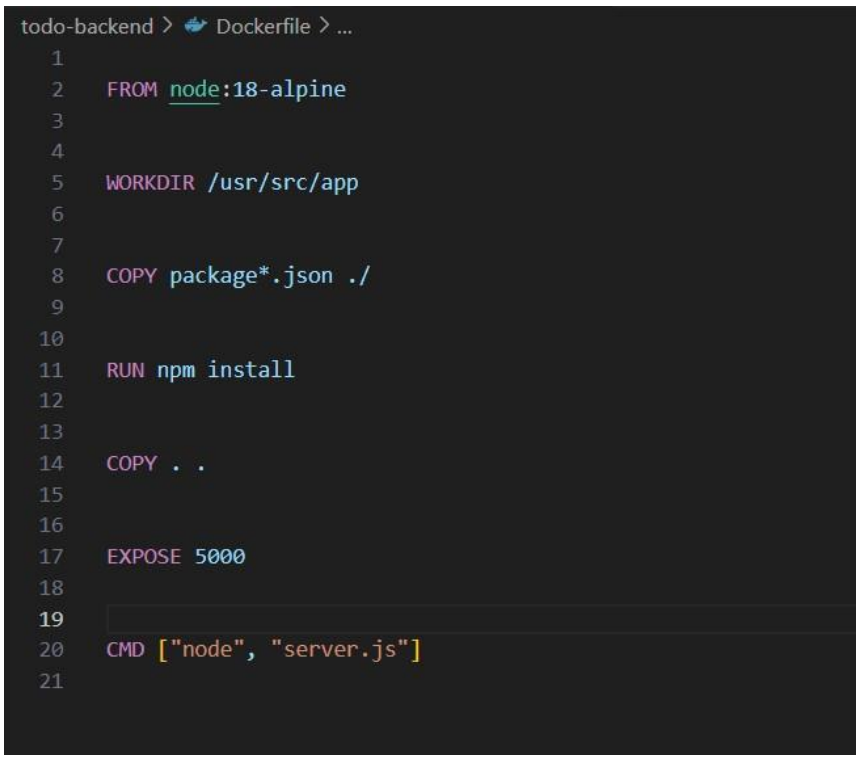
1. **Define a multi-service application consisting of at least three services (e.g., a web app, a database, and a caching service) in a docker-compose.yml file.**
2. **Configure the services with appropriate environment variables and networking settings.**
3. **Build and run the Docker Compose application.**
4. **Test the connectivity between the services and the functionality of the application.**

1. Prerequisites:

- **Ensure Docker is installed on your system.**
- **Have a web application ready or create a simple one.**

2. Dockerfiles:

Backend Dockerfile:



Frontend Dockerfile:

3. Docker Compose Configuration (docker-compose.yml):

- **version: '3.8':** Specifies the Docker Compose file format version.
- **services:**
    - **frontend:**
        - **image:** Specifies the Docker image for the frontend service.
        - **build:** Specifies the build context and Dockerfile for the frontend service.
        - **ports:** Maps port 3000 on the host to port 3000 in the container.
    - **backend:**
        - **image:** Specifies the Docker image for the backend service.
        - **build:** Specifies the build context and Dockerfile for the backend service.
        - **ports:** Maps port 5000 on the host to port 5000 in the container.
        - **depends_on:** Ensures the backend service starts only after the 'mongo' service.
        - **networks:** Specifies the 'todo-network' for communication between services.
    - **mongo:**
        - **image:** Specifies the Docker image for the MongoDB service.
        - **ports:** Maps port 27017 on the host to port 27017 in the container.
        - **networks:** Specifies the 'todo-network' for communication between services.
- **networks:**
    - **todo-network:** Defines a custom Docker network for communication between services.

4. Build and Run Docker Compose Application:

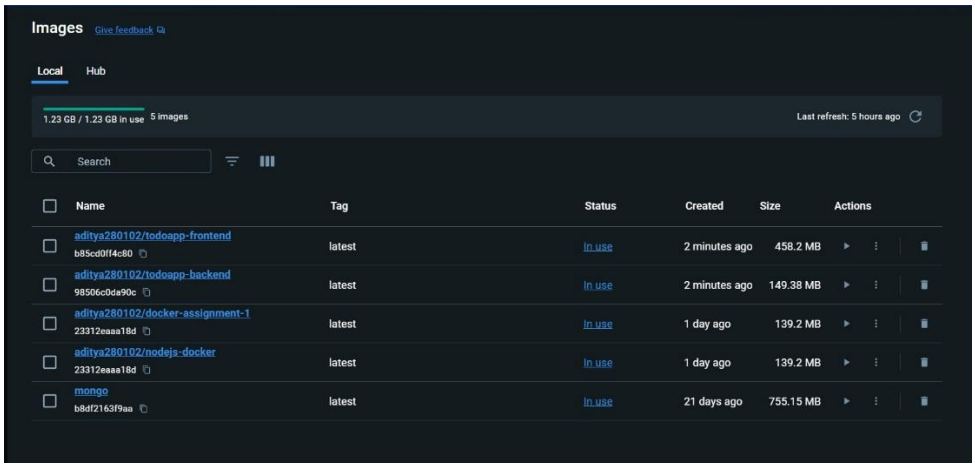
- **Build and run the Docker Compose application:**

  “docker-compose up -d –build”
- **Explanation:**
    - **-d**: Run containers in the background.
    - **--build**: Build images before starting containers.

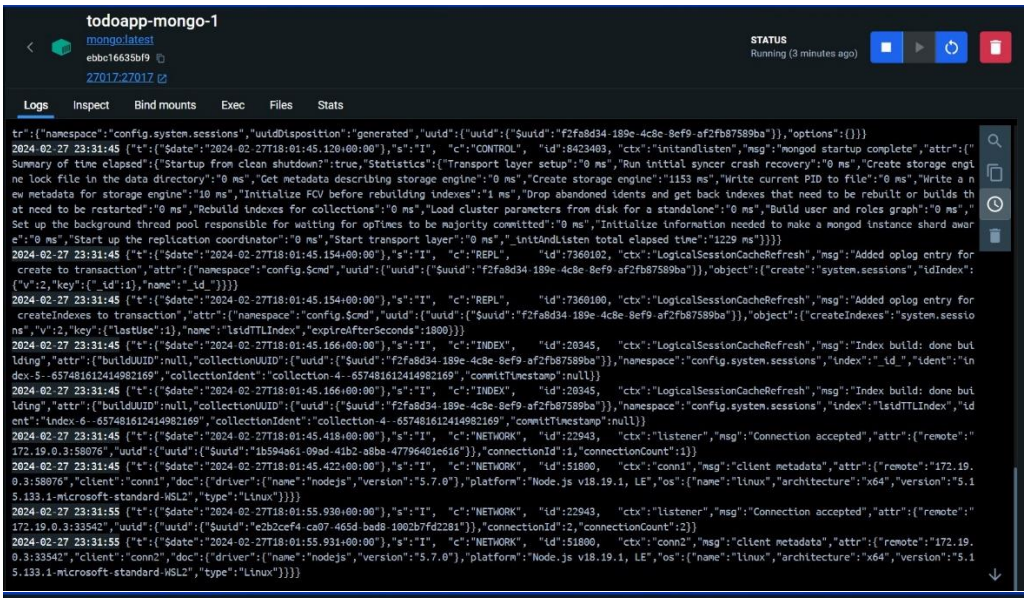
```
PS C:\Users\HP\Desktop\Todo app> docker compose up -d --build
[+] Running 9/9
 ✓ mongo 8 layers [⣿⣿⣿⣿⣿⣿⣿⣿]      0B/0B      Pulled                                                                                    7.8s
   ✓ 01007420e9b0 Already exists                                                                                                       0.0s
   ✓ bc3bec6a423e Already exists                                                                                                       0.0s
   ✓ c5db81b694a8 Already exists                                                                                                       0.0s
   ✓ 427a1a117df0 Already exists                                                                                                       0.0s
   ✓ dfb180c9e7b5 Already exists                                                                                                       0.0s
   ✓ 92e6f08e133c Already exists                                                                                                       0.0s
   ✓ 374f042f3159 Already exists                                                                                                       0.0s
   ✓ 73549bb43006 Already exists                                                                                                       0.0s
2024/02/27 23:31:40 http2: server: error reading preface from client //./pipe/docker_engine: file has already been closed
[+] Building 0.0s (0/0)  docker:default
2024/02/27 23:31:40 http2: server: error reading preface from client //./pipe/docker_engine: file has already been closed
[+] Building 1.3s (17/17) FINISHED                                                                                          docker:default
 => [frontend internal] load build definition from Dockerfile                                                                        0.0s
 => => transferring dockerfile: 193B                                                                                                 0.0s
 => [backend internal] load build definition from Dockerfile                                                                         0.0s
 => => transferring dockerfile: 198B                                                                                                 0.0s
 => [backend internal] load metadata for docker.io/library/node:18-alpine                                                            1.1s
 => [frontend internal] load .dockerignore                                                                                           0.0s
 => => transferring context: 2B                                                                                                      0.0s
 => [backend internal] load .dockerignore                                                                                            0.0s
 => => transferring context: 2B                                                                                                      0.0s
 => [frontend internal] load build context                                                                                           0.0s
 => => transferring context: 1.12kB                                                                                                  0.0s
 => [backend 1/5] FROM docker.io/library/node:18-alpine@sha256:ca9f6cb0466f9638e59e0c249d335a07c867cd50c429b5c7830dda1bed584649      0.0s
 => [backend internal] load build context                                                                                            0.0s
 => => transferring context: 191B                                                                                                    0.0s
 => CACHED [frontend 2/5] WORKDIR /usr/src/app                                                                                       0.0s
 => CACHED [backend 3/5] COPY package*.json ./                                                                                       0.0s
 => CACHED [backend 4/5] RUN npm install                                                                                             0.0s
 => CACHED [backend 5/5] COPY . .                                                                                                    0.0s
 => CACHED [frontend 3/5] COPY package*.json ./                                                                                      0.0s
 => CACHED [frontend 4/5] RUN npm install                                                                                            0.0s
 => CACHED [frontend 5/5] COPY . .                                                                                                   0.0s
 => [backend] exporting to image                                                                                                     0.0s
 => => exporting layers                                                                                                              0.0s
 => => writing image sha256:98506c0da90c4a0a3262dbdd9e58dc1e1b9f33d4c270f93d2669d24ab4d1c68f                                         0.0s
 => => naming to docker.io/aditya280102/todoapp-backend                                                                              0.0s
 => [frontend] exporting to image                                                                                                    0.0s
 => => exporting layers                                                                                                              0.0s
 => => writing image sha256:b85cd0ff4c80e5c15e4c210473f494ab53bd4baa0c0b6ab0e8352bcca11c8a41                                         0.0s
 => => naming to docker.io/aditya280102/todoapp-frontend                                                                             0.0s
[+] Running 3/5
```

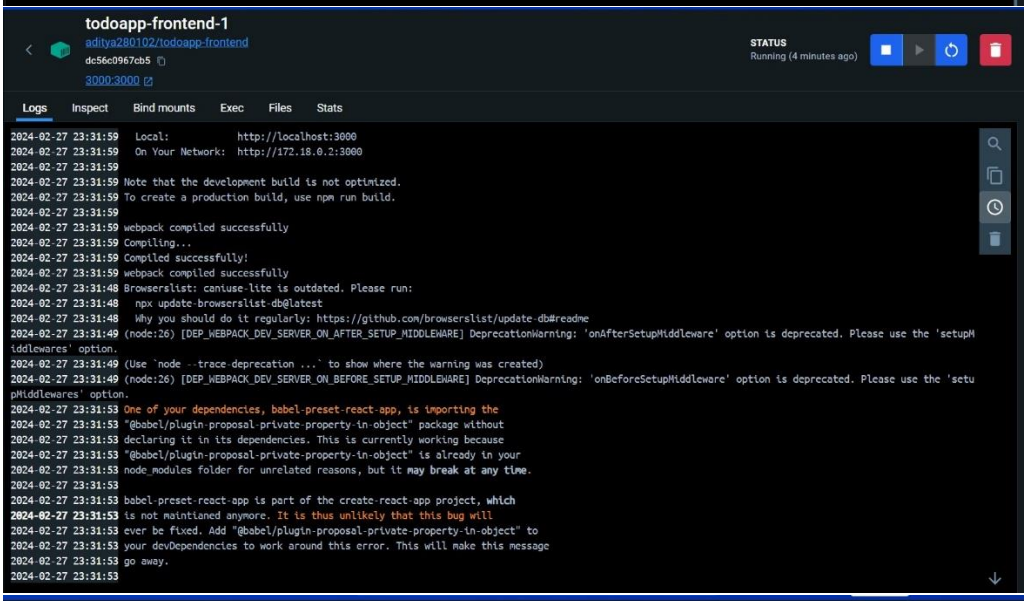
```
 => CACHED [backend 3/5] COPY package*.json ./                                                                                       0.0s
 => CACHED [backend 4/5] RUN npm install                                                                                             0.0s
 => CACHED [backend 5/5] COPY . .                                                                                                    0.0s
 => CACHED [frontend 3/5] COPY package*.json ./                                                                                      0.0s
 => CACHED [frontend 4/5] RUN npm install                                                                                            0.0s
 => CACHED [frontend 5/5] COPY . .                                                                                                   0.0s
 => [backend] exporting to image                                                                                                     0.0s
 => => exporting layers                                                                                                              0.0s
 => => writing image sha256:98506c0da90c4a0a3262dbdd9e58dc1e1b9f33d4c270f93d2669d24ab4d1c68f                                         0.0s
 => => naming to docker.io/aditya280102/todoapp-backend                                                                              0.0s
 => [frontend] exporting to image                                                                                                    0.0s
 => => exporting layers                                                                                                              0.0s
 => => writing image sha256:b85cd0ff4c80e5c15e4c210473f494ab53bd4baa0c0b6ab0e8352bcca11c8a41                                         0.0s
 => => naming to docker.io/aditya280102/todoapp-frontend                                                                             0.0s
[+] Running 3/5
 - Network todoapp_default       Created                                                                                             2.6s
 - Network todoapp_todo-network  Created                                                                                             2.4s
 ✓ Container todoapp-mongo-1     Started                                                                                             1.9s
 ✓ Container todoapp-frontend-1  Started                                                                                             1.9s
 ✓ Container todoapp-backend-1   Started                                                                                             2.0s
PS C:\Users\HP\Desktop\Todo app>
```

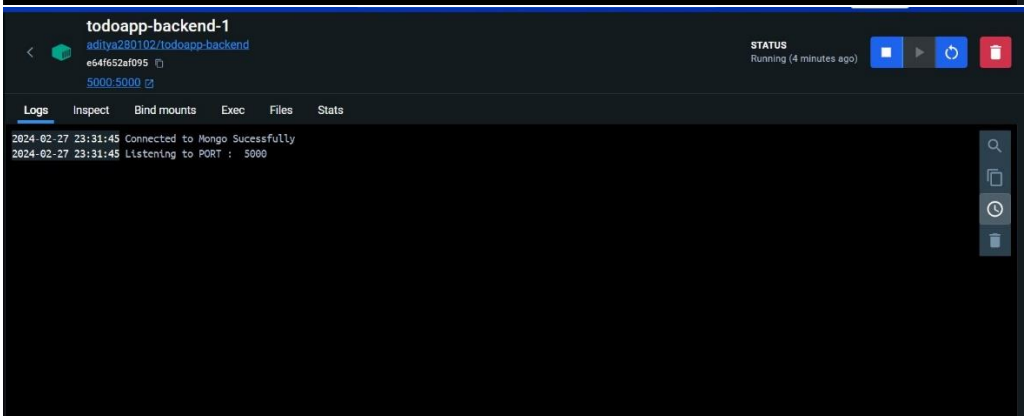
Result:



Logs:


todoapp-mongo-1
mongo:latest
ebbc16635bf9
27017:27017
STATUS
Running (3 minutes ago)
Logs
Inspect
Bind mounts
Exec
Files
Stats
tr":{"namespace":"config.system.sessions","uuidDisposition":"generated","uuid":{"uuid":{"$uuid":"f2fa8d34-189e-4c8e-8ef9-af2fb87589ba"}},"options":{}}}
2024-02-27 23:31:45 {"t":{"$date":"2024-02-27T18:01:45.120+00:00"},"s":"I", "c":"CONTROL", "id":8423403, "ctx":"initandlisten","msg":"mongod startup complete","attr":{"
Summary of time elapsed":{"Startup from clean shutdown?":true,"Statistics":{"Transport layer setup":"0 ms","Run initial syncer crash recovery":"0 ms","Create storage engi
ne lock file in the data directory":"0 ms","Get metadata describing storage engine":"0 ms","Create storage engine":"1153 ms","Write current PID to file":"0 ms","Write a n
ew metadata for storage engine":"10 ms","Initialize FCV before rebuilding indexes":"1 ms","Drop abandoned idents and get back indexes that need to be rebuilt or builds th
at need to be restarted":"0 ms","Rebuild indexes for collections":"0 ms","Load cluster parameters from disk for a standalone":"0 ms","Build user and roles graph":"0 ms","
Set up the background thread pool responsible for waiting for opTimes to be majority committed":"0 ms","Initialize information needed to make a mongod instance shard awar
e":"0 ms","Start up the replication coordinator":"0 ms","Start transport layer":"0 ms","_initAndListen total elapsed time":"1229 ms"}}}}
2024-02-27 23:31:45 {"t":{"$date":"2024-02-27T18:01:45.154+00:00"},"s":"I", "c":"REPL", "id":7360102, "ctx":"LogicalSessionCacheRefresh","msg":"Added oplog entry for
create to transaction","attr":{"namespace":"config.$cmd","uuid":{"uuid":{"$uuid":"f2fa8d34-189e-4c8e-8ef9-af2fb87589ba"}},"object":{"create":"system.sessions","idIndex":
{"v":2,"key":{"_id":1},"name":"_id_"}}}}
2024-02-27 23:31:45 {"t":{"$date":"2024-02-27T18:01:45.154+00:00"},"s":"I", "c":"REPL", "id":7360100, "ctx":"LogicalSessionCacheRefresh","msg":"Added oplog entry for
createIndexes to transaction","attr":{"namespace":"config.$cmd","uuid":{"uuid":{"$uuid":"f2fa8d34-189e-4c8e-8ef9-af2fb87589ba"}},"object":{"createIndexes":"system.sessio
ns","v":2,"key":{"lastUse":1},"name":"lsidTTLIndex","expireAfterSeconds":1800}}}
2024-02-27 23:31:45 {"t":{"$date":"2024-02-27T18:01:45.166+00:00"},"s":"I", "c":"INDEX", "id":20345, "ctx":"LogicalSessionCacheRefresh","msg":"Index build: done bui
lding","attr":{"buildUUID":null,"collectionUUID":{"uuid":{"$uuid":"f2fa8d34-189e-4c8e-8ef9-af2fb87589ba"}},"namespace":"config.system.sessions","index":"_id_","ident":"in
dex-5--6574816124149821 69","collectionIdent":"collection-4--6574816124149821 69","commitTimestamp":null}}
2024-02-27 23:31:45 {"t":{"$date":"2024-02-27T18:01:45.166+00:00"},"s":"I", "c":"INDEX", "id":20345, "ctx":"LogicalSessionCacheRefresh","msg":"Index build: done bui
lding","attr":{"buildUUID":null,"collectionUUID":{"uuid":{"$uuid":"f2fa8d34-189e-4c8e-8ef9-af2fb87589ba"}},"namespace":"config.system.sessions","index":"lsidTTLIndex","id
ent":"index-6--6574816124149821 69","collectionIdent":"collection-4--6574816124149821 69","commitTimestamp":null}}
2024-02-27 23:31:45 {"t":{"$date":"2024-02-27T18:01:45.418+00:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"Connection accepted","attr":{"remote":"
172.19.0.3:58076","uuid":{"uuid":{"$uuid":"1b594a61-09ad-41b2-a8ba-47796401e616"}},"connectionId":1,"connectionCount":1}}
2024-02-27 23:31:45 {"t":{"$date":"2024-02-27T18:01:45.422+00:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn1","msg":"client metadata","attr":{"remote":"172.19.
0.3:58076","client":"conn1","doc":{"driver":{"name":"nodejs","version":"5.7.0"},"platform":"Node.js v18.19.1, LE","os":{"name":"linux","architecture":"x64","version":"5.1
5.133.1-microsoft-standard-WSL2","type":"Linux"}}}}
2024-02-27 23:31:55 {"t":{"$date":"2024-02-27T18:01:55.930+00:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"Connection accepted","attr":{"remote":"
172.19.0.3:33542","uuid":{"uuid":{"$uuid":"e2b2cef4-ca07-465d-bad8-1002b7fd2281"}},"connectionId":2,"connectionCount":2}}
2024-02-27 23:31:55 {"t":{"$date":"2024-02-27T18:01:55.931+00:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn2","msg":"client metadata","attr":{"remote":"172.19.
0.3:33542","client":"conn2","doc":{"driver":{"name":"nodejs","version":"5.7.0"},"platform":"Node.js v18.19.1, LE","os":{"name":"linux","architecture":"x64","version":"5.1
5.133.1-microsoft-standard-WSL2","type":"Linux"}}}}

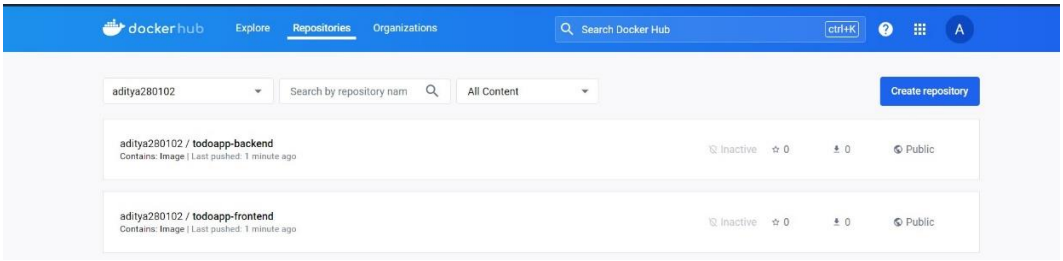

todoapp-frontend-1
aditya280102/todoapp-frontend
dc56c0967cb5
3000:3000
STATUS
Running (4 minutes ago)
Logs
Inspect
Bind mounts
Exec
Files
Stats
2024-02-27 23:31:59 Local: http://localhost:3000
2024-02-27 23:31:59 On Your Network: http://172.18.0.2:3000
2024-02-27 23:31:59
2024-02-27 23:31:59 Note that the development build is not optimized.
2024-02-27 23:31:59 To create a production build, use npm run build.
2024-02-27 23:31:59
2024-02-27 23:31:59 webpack compiled successfully
2024-02-27 23:31:59 Compiling...
2024-02-27 23:31:59 Compiled successfully!
2024-02-27 23:31:59 webpack compiled successfully
2024-02-27 23:31:48 Browserslist: caniuse-lite is outdated. Please run:
2024-02-27 23:31:48 npx update-browserslist-db@latest
2024-02-27 23:31:48 Why you should do it regularly: https://github.com/browserslist/update-db#readme
2024-02-27 23:31:49 (node:26) [DEP_WEBPACK_DEV_SERVER_ON_AFTER_SETUP_MIDDLEWARE] DeprecationWarning: 'onAfterSetupMiddleware' option is deprecated. Please use the 'setupM
iddlewares' option.
2024-02-27 23:31:49 (Use `node --trace-deprecation ...` to show where the warning was created)
2024-02-27 23:31:49 (node:26) [DEP_WEBPACK_DEV_SERVER_ON_BEFORE_SETUP_MIDDLEWARE] DeprecationWarning: 'onBeforeSetupMiddleware' option is deprecated. Please use the 'setu
pMiddlewares' option.
2024-02-27 23:31:53 One of your dependencies, babel-preset-react-app, is importing the
2024-02-27 23:31:53 "@babel/plugin-proposal-private-property-in-object" package without
2024-02-27 23:31:53 declaring it in its dependencies. This is currently working because
2024-02-27 23:31:53 "@babel/plugin-proposal-private-property-in-object" is already in your
2024-02-27 23:31:53 node_modules folder for unrelated reasons, but it may break at any time.
2024-02-27 23:31:53
2024-02-27 23:31:53 babel-preset-react-app is part of the create-react-app project, which
2024-02-27 23:31:53 is not maintained anymore. It is thus unlikely that this bug will
2024-02-27 23:31:53 ever be fixed. Add "@babel/plugin-proposal-private-property-in-object" to
2024-02-27 23:31:53 your devDependencies to work around this error. This will make this message
2024-02-27 23:31:53 go away.
2024-02-27 23:31:53


todoapp-backend-1
aditya280102/todoapp-backend
e64f652af095
5000:5000
STATUS
Running (4 minutes ago)
Logs
Inspect
Bind mounts
Exec
Files
Stats
2024-02-27 23:31:45 Connected to Mongo Sucessfully
2024-02-27 23:31:45 Listening to PORT : 5000

- **Now, Push it to the Docker hub:**
  Login docker (“docker login”)

  Then, run “docker compose push”

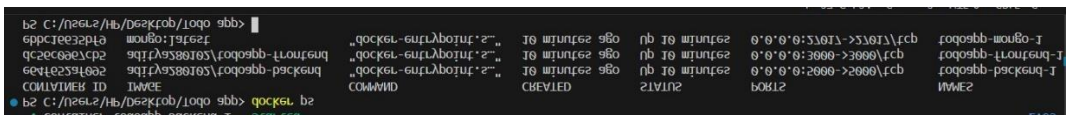
  Result:



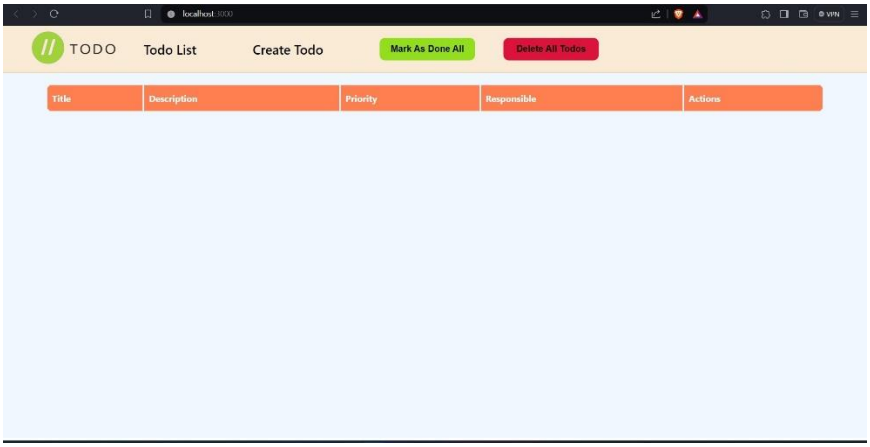
5. Testing:

- Ensure all services are up and running:

  “docker-compose ps”

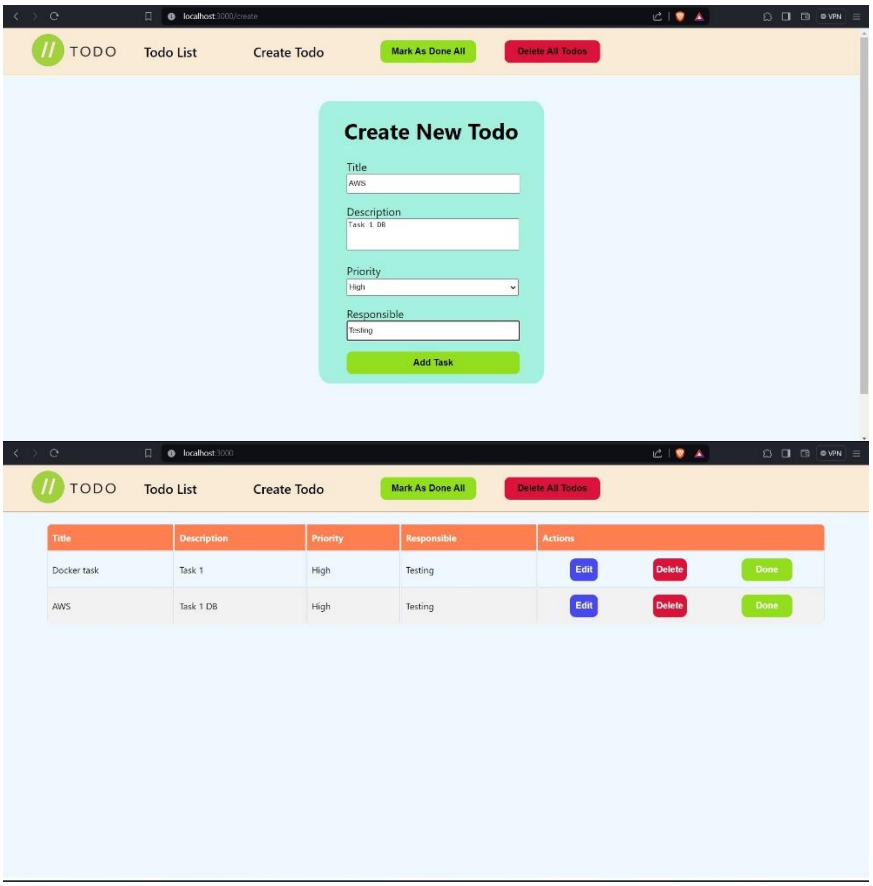


- Access the frontend at **http://localhost:3000** and test the web application's functionality.

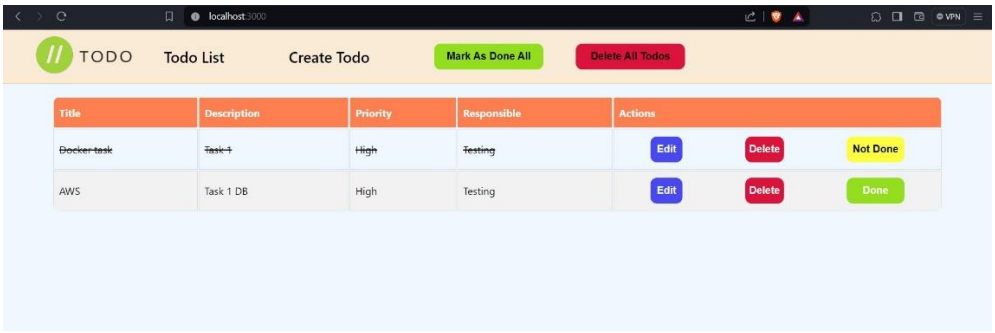


Some functionalities of the app:

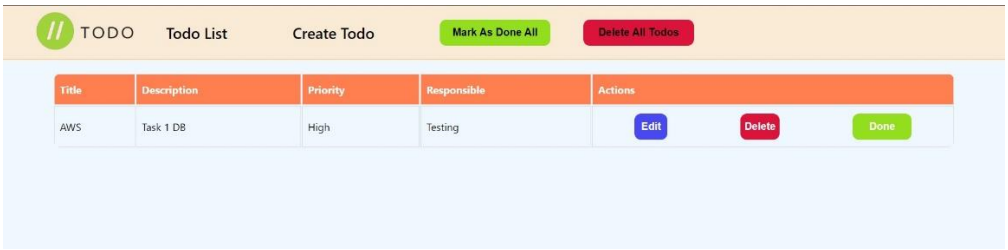
Create New Todo:



We can mark the Todo as done:



We can delete the Todo:



- Check the backend API at **http://localhost:5000**.

6. Cleanup:

- Stop and remove the Docker Compose application:

"docker-compose down"

7. Conclusion:

- The multi-service application is now orchestrated using Docker Compose.
- The application components (frontend, backend, and MongoDB) communicate seamlessly within the custom network.
- Follow the cleanup steps when done testing or deploying the application.