

# WIF3005 Software Maintenance and Evolution

Mohamad Sulaiman bin Mohd Hashim

17096959/3

## Alternative Assessment

### Question 2

Repository: <https://github.com/sulaimanmhashim/javascript-tetris>

---

2. Migrate an existing software system into a containerized environment using Docker. This includes creating a Dockerfile and setting up the necessary configuration to run your application in a Docker container.

2.1. Select your Project: Option 2 : Use the sample repository provided in Figure 1.

2.1.1. Forked the <https://github.com/jakesgordon/javascript-tetris> into <https://github.com/sulaimanmhashim/javascript-tetris>

2.2. Create a Dockerfile:

2.2.1. Created a Dockerfile in the root directory of the project.

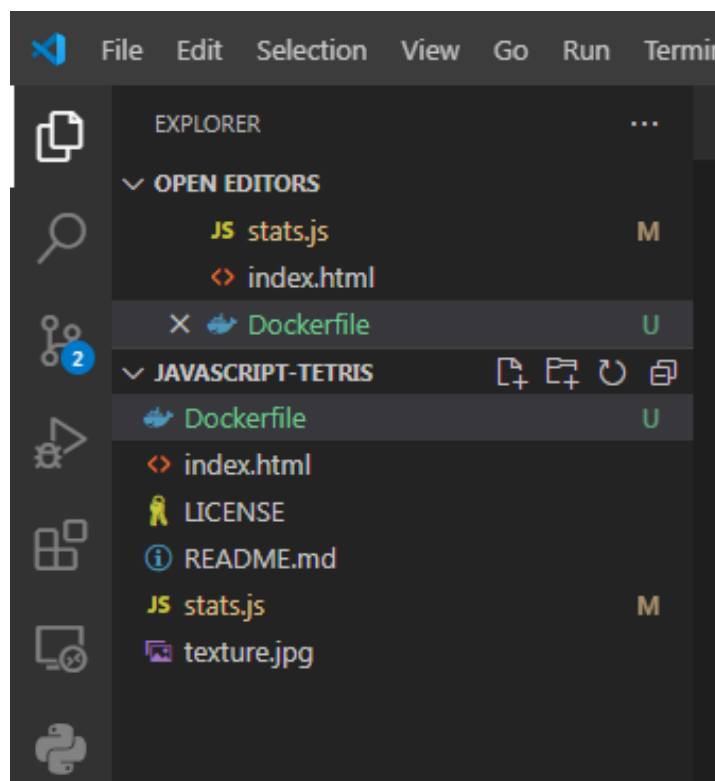


Image 2.2.1.: Created the Dockerfile in the root directory.

2.2.2. Ensured the Dockerfile includes all required (Base image, working directory, dependencies, port, and run command).

2.2.2.1. Code Snippet:

```
# Use the official nginx image as the base
FROM nginx:latest

# Set the working directory
WORKDIR /usr/share/nginx/html

# Copy project files to the container
COPY . .

# Change nginx configuration to listen on port 5000
RUN sed -i 's/listen      80;/listen      5000;/g'
/etc/nginx/conf.d/default.conf

# Expose port 5000 to the host
EXPOSE 5000

# Run nginx in the foreground
CMD ["nginx", "-g", "daemon off;"]
```

## 2.3. Build and test the the Docker Image:

### 2.3.1. Built using this sample command: docker build -t javascript-tetris-html .

```
Sulaiman@Goofstation MINGW64 /d/sulaiman/javascript-tetris (master)
● $ docker build -t javascript-tetris-html .
○ [+] Building 1.0s (9/9) FINISHED
  => [internal] load build definition from Dockerfile
  => => transferring dockerfile: 469B
  => [internal] load metadata for docker.io/library/nginx:latest
  => [internal] load .dockerignore
  => => transferring context: 2B
  => [1/4] FROM docker.io/library/nginx:latest@sha256:0a399eb16751829e1af26fea27b20c3ec28d7ab1fb72182879dcae1cca21206a
  => => resolve docker.io/library/nginx:latest@sha256:0a399eb16751829e1af26fea27b20c3ec28d7ab1fb72182879dcae1cca21206a
  => [internal] load build context
  => => transferring context: 2.95kB
  => CACHED [2/4] WORKDIR /usr/share/nginx/html
  => CACHED [3/4] COPY . .
  => CACHED [4/4] RUN sed -i 's/listen      80;/listen      5000;/g' /etc/nginx/conf.d/default.conf
  => exporting to image
  => => exporting layers
  => => exporting manifest sha256:5c315e169dc0c7e52c8ae5f498bf0131784eb6ca0408a13f5ac130910dc34c08
  => => exporting config sha256:f91831aa0e93106b8f199a4d28136b6e388bdb3b1ea16c9063b109487f2a68ba
  => => exporting attestation manifest sha256:e6bfeadb3a63bb89aa3c833575374e4315341a5790544478c81827db00a36b9a
  => => exporting manifest list sha256:6251b2ab52b290e84323ef85f29934a3504ad200de652dcaa3f50c1a55cce36c
  => => naming to docker.io/library/javascript-tetris-html:latest
  => => unpacking to docker.io/library/javascript-tetris-html:latest

Sulaiman@Goofstation MINGW64 /d/sulaiman/javascript-tetris (master)
$
```

Image 2.3.1.: Successful build of the Dockerfile.

### 2.3.2. Run the Docker container using this sample command: docker run -p 5000:5000 javascript-tetris-html

```
Sulaiman@Goofstation MINGW64 /d/sulaiman/javascript-tetris (master)
$ docker run -p 5000:5000 javascript-tetris-html
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: /etc/nginx/conf.d/default.conf differs from the packaged version
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2025/01/22 04:31:22 [notice] 1#1: using the "epoll" event method
2025/01/22 04:31:22 [notice] 1#1: nginx/1.27.3
2025/01/22 04:31:22 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)
2025/01/22 04:31:22 [notice] 1#1: OS: Linux 5.15.167.4-microsoft-standard-WSL2
2025/01/22 04:31:22 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2025/01/22 04:31:22 [notice] 1#1: start worker processes
2025/01/22 04:31:22 [notice] 1#1: start worker process 28
2025/01/22 04:31:22 [notice] 1#1: start worker process 29
2025/01/22 04:31:22 [notice] 1#1: start worker process 30
2025/01/22 04:31:22 [notice] 1#1: start worker process 31
2025/01/22 04:31:22 [notice] 1#1: start worker process 32
2025/01/22 04:31:22 [notice] 1#1: start worker process 33
2025/01/22 04:31:22 [notice] 1#1: start worker process 34
2025/01/22 04:31:22 [notice] 1#1: start worker process 35
2025/01/22 04:31:22 [notice] 1#1: start worker process 36
2025/01/22 04:31:22 [notice] 1#1: start worker process 37
2025/01/22 04:31:22 [notice] 1#1: start worker process 38
2025/01/22 04:31:22 [notice] 1#1: start worker process 39
```

Image 2.3.2.: Successful run of the Docker container.

2.3.3. Verified the application is accessible in a browser.

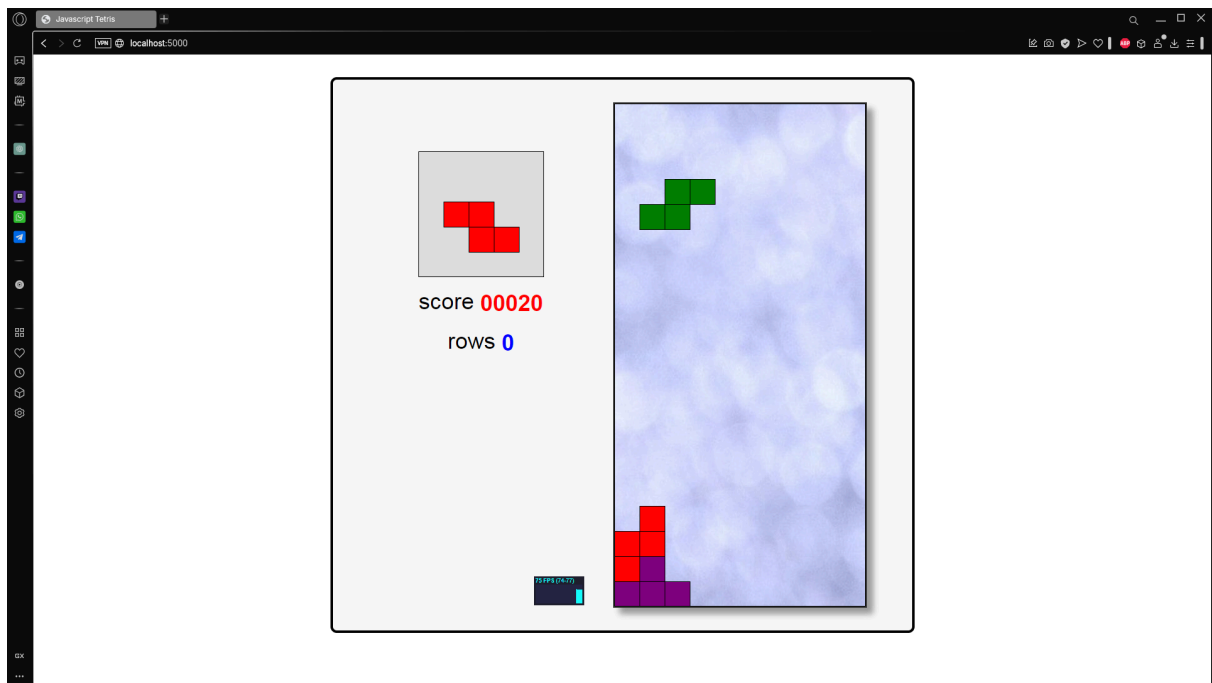


Image 2.3.3.: Opened <http://localhost:5000> on default browser.