

**UNIVERSITI
MALAYA**

**WIF3005 SOFTWARE MAINTENANCE AND
EVOLUTION**

SEMESTER 1 2025/2026

ALTERNATIVE ASSESSMENT

**Using FYP2 Project: Smart Desktop Buddies - For Good Mental
Health and Effective Learning**

Name: Nurmohamad Aiman bin Alias

Matric Number: U2100973/2

Lecturer: DR. NUR NASUHA BINTI MOHD DAUD

The assignment uses my final year project titled smart desktop buddies, with the link to the repo [here](#) at the docker-assignment branch of the repo.

Screenshots of proof

1. Folder Structure

```
(venv) PS C:\Users\Aiman\Desktop\Semester 7 FCSIT\WIA3003 AP2\smartdesktopbuddies\watermelon\smart-desktop-buddie
s> ls

Directory: C:\Users\Aiman\Desktop\Semester 7 FCSIT\WIA3003
AP2\smartdesktopbuddies\watermelon\smart-desktop-buddies

Mode                LastWriteTime         Length Name
----                -
d-----          18/1/2026   8:46 PM             backend
d-----          22/1/2026   2:18 AM             BuildTest
d-----          22/1/2026   2:19 AM             Dockerfile
d-----          18/1/2026   8:41 PM             frontend
d-----          20/12/2025   8:19 PM             venv
-a-----          20/12/2025   8:19 PM              68 .gitattributes
-a-----          15/1/2026    3:44 AM             863 .gitignore
-a-----          18/1/2026   12:18 AM             9588 README.md

(venv) PS C:\Users\Aiman\Desktop\Semester 7 FCSIT\WIA3003 AP2\smartdesktopbuddies\watermelon\smart-desktop-buddie
s>
```

Figure 1.0: Required Folder Structure

Screenshot shows the Dockerfile and BuildTest folders created in the project root directory as required by the assignment.

```
(venv) PS C:\Users\Aiman\Desktop\Semester 7 FCSIT\WIA3003 AP2\smartdesktopbuddies\watermelon\smart-desktop-buddie
s> Dockerfile> ls


Directory: C:\Users\Aiman\Desktop\Semester 7 FCSIT\WIA3003
AP2\smartdesktopbuddies\watermelon\smart-desktop-buddies\Dockerfile

Mode                LastWriteTime         Length Name
----                -
-a-----          22/1/2026    2:24 AM             1063 Dockerfile
```

Figure 1.1: Dockerfile file inside Dockerfile folder

Screenshot shows the Dockerfile is inside the Dockerfile folder of the project root directory

2.0 Dockerfile Content

A screenshot of a code editor window titled 'Dockerfile U'. The editor shows a Dockerfile with 28 lines of code. The code includes comments for repository link, base image, working directory, requirements installation, dependencies, port exposure, environment variables, and the final run command. The code is as follows:

```
1 # Repository link: https://github.com/Aiman-alias/smart-desktop-buddies/tree/docker-assignment
2 # Base image with Python 3.11
3 FROM python:3.11-slim
4
5 # Set my working directory inside container
6 WORKDIR /app
7
8 # Copy requirements file first (for better caching later on i think)
9 COPY backend/requirements.txt /app/requirements.txt
10
11 # to install Python dependencies (using pip)
12 RUN pip install --no-cache-dir -r requirements.txt
13
14 # this to copy entire backend directory
15 COPY backend/ /app/backend/
16
17 # Set working directory to Django project root (where manage.py is)
18 WORKDIR /app/backend/core
19
20 # Expose port 8000 (i use this because its the Django default port)
21 EXPOSE 8000
22
23 # Set environment variables (using defaults if not provided because i don't want to change the default values)
24 ENV SECRET_KEY=django-insecure-docker-secret-key-for-testing
25 ENV DEBUG=True
26
27 # Run database migrations and start server (i use this because its the command i used to start the server)
28 CMD python manage.py migrate && python manage.py runserver 0.0.0.0:8000
```

Figure 2.0 The content of the Dockerfile

Screenshot shows the complete Dockerfile with all required elements: base image, working directory, dependencies installation, port exposure, and run command. The **repository link comment** is visible at the very top.

3.0 Docker Build Processes

```
Problems Output Debug Console Terminal Ports
PS C:\Users\Aiman\Desktop\Semester 7 FCSIT\WIA3003 AP2\smartdesktopbuddies\watermelon\smart-desktop-buddies> docker
er build -t smart-desktop-buddies -f Dockerfile/Dockerfile .
[+] Building 54.0s (11/11) FINISHED
er:desktop-linux
=> [internal] load build definition from Dockerfile
0.0s
=> => transferring dockerfile: 1.10kB
0.0s
=> [internal] load metadata for docker.io/library/python:3.11-slim
1.8s
=> [internal] load .dockerignore
0.0s
=> => transferring context: 2B
0.0s
=> [1/6] FROM docker.io/library/python:3.11-slim@sha256:5be45dbade29bebd6886af6b438fd7e0b4eb7b611f39ba62b430263f
82de36d2 0.0s
=> => resolve docker.io/library/python:3.11-slim@sha256:5be45dbade29bebd6886af6b438fd7e0b4eb7b611f39ba62b430263f
82de36d2 0.0s
=> [internal] load build context
0.9s
=> => transferring context: 1.33MB
0.9s
=> CACHED [2/6] WORKDIR /app
0.0s
=> [3/6] COPY backend/requirements.txt /app/requirements.txt
0.0s
=> [4/6] RUN pip install --no-cache-dir -r requirements.txt
30.8s
=> [5/6] COPY backend/ /app/backend/
3.8s
=> [6/6] WORKDIR /app/backend/core
0.2s
```

Figure 3.0: Docker Build Command (1/2)

```
Problems Output Debug Console Terminal Ports
=> [6/6] WORKDIR /app/backend/core
0.2s
=> exporting to image
16.2s
=> => exporting layers
6.0s
=> => exporting manifest sha256:3d8e51514447fd829616df1446b8c38ee969823754c9e5fe6d5f84d3393269cd
0.0s
=> => exporting config sha256:b0cc54ae87c5e30e45e724e82332c5b31f3d1e7d3faa2c685e2433982bf7953f
0.0s
=> => exporting attestation manifest sha256:3f61c95305b5c4e140936dba5e92adb11512ca5447fe0bb563f5d7a051334e3a
0.0s
=> => exporting manifest list sha256:ccce83c29c9f11966899da4326246d9fac7e5ed182c00da260d0d66c32a1b141
0.0s
=> => naming to docker.io/library/smart-desktop-buddies:latest
0.0s
=> => unpacking to docker.io/library/smart-desktop-buddies:latest
10.1s

2 warnings found (use docker --debug to expand):
- JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (
line 28)
- SecretsUsedInArgOrEnv: Do not use ARG or ENV instructions for sensitive data (ENV "SECRET_KEY") (line 24)
```

Figure 3.1: Docker Build Command (2/2)

Screenshots show the docker build command **docker build -t smart-desktop-buddies -f Dockerfile/Dockerfile .** being executed with the correct parameters: image name (-t), Dockerfile path (-f), and build context (.) and **was successful (11/11) FINISHED**. Both warnings are informational recommendations, not errors. The build completed successfully, and the application functions correctly as screenshotted.

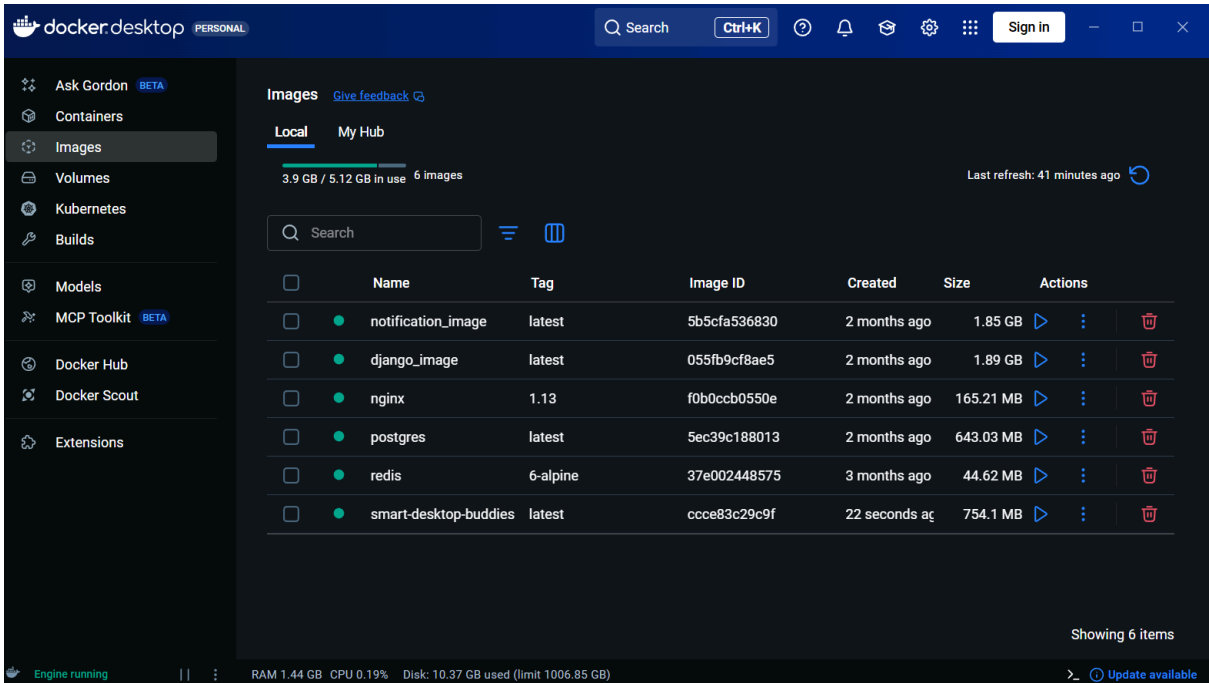


Figure 3.2: docker-desktop local images

Screenshot shows the smart-desktop-buddies local image was created (at the very bottom of the list) along with the tag, image ID, date created and size. (Ignore all the other images that were from other projects.)

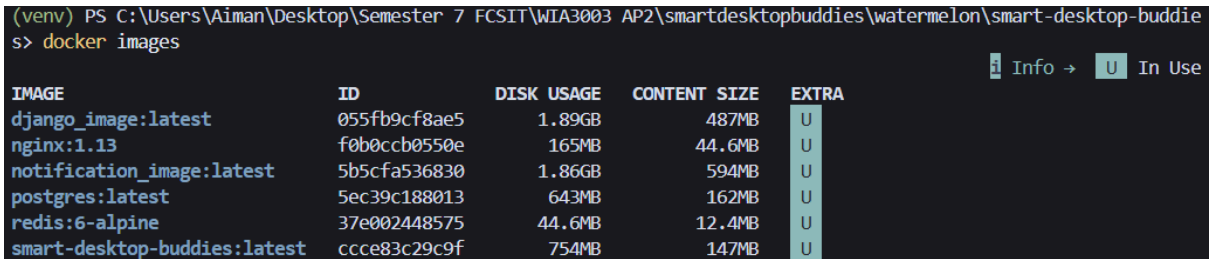
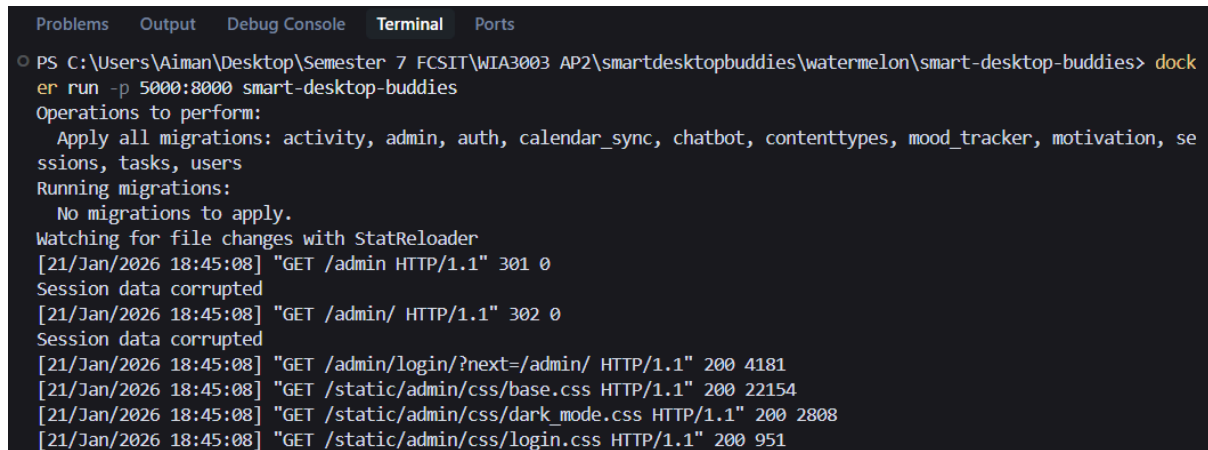


Figure 3.3: docker images command in the powershell terminal

Screenshot shows the smart-desktop-buddies image is in fact created by looking at the terminal output of **docker images**.

4.0 Docker Run Processes



```
Problems Output Debug Console Terminal Ports
PS C:\Users\Aiman\Desktop\Semester 7 FCSIT\WIA3003 AP2\smartdesktopbuddies\watermelon\smart-desktop-buddies> docker
er run -p 5000:8000 smart-desktop-buddies
Operations to perform:
  Apply all migrations: activity, admin, auth, calendar_sync, chatbot, contenttypes, mood_tracker, motivation, se
ssions, tasks, users
Running migrations:
  No migrations to apply.
Watching for file changes with StatReloader
[21/Jan/2026 18:45:08] "GET /admin HTTP/1.1" 301 0
Session data corrupted
[21/Jan/2026 18:45:08] "GET /admin/ HTTP/1.1" 302 0
Session data corrupted
[21/Jan/2026 18:45:08] "GET /admin/login/?next=/admin/ HTTP/1.1" 200 4181
[21/Jan/2026 18:45:08] "GET /static/admin/css/base.css HTTP/1.1" 200 22154
[21/Jan/2026 18:45:08] "GET /static/admin/css/dark_mode.css HTTP/1.1" 200 2808
[21/Jan/2026 18:45:08] "GET /static/admin/css/login.css HTTP/1.1" 200 951
```

Figure 4.0: Docker Run command

Screenshot shows the command **docker run -p 5000:8000 smart-desktop-buddies** was run and successful along with the logs of the backend (as proof that it works). Also shows Django running database migrations automatically when the container starts. All required apps are listed and migrations are applied successfully.

5.0 Browser Screenshots

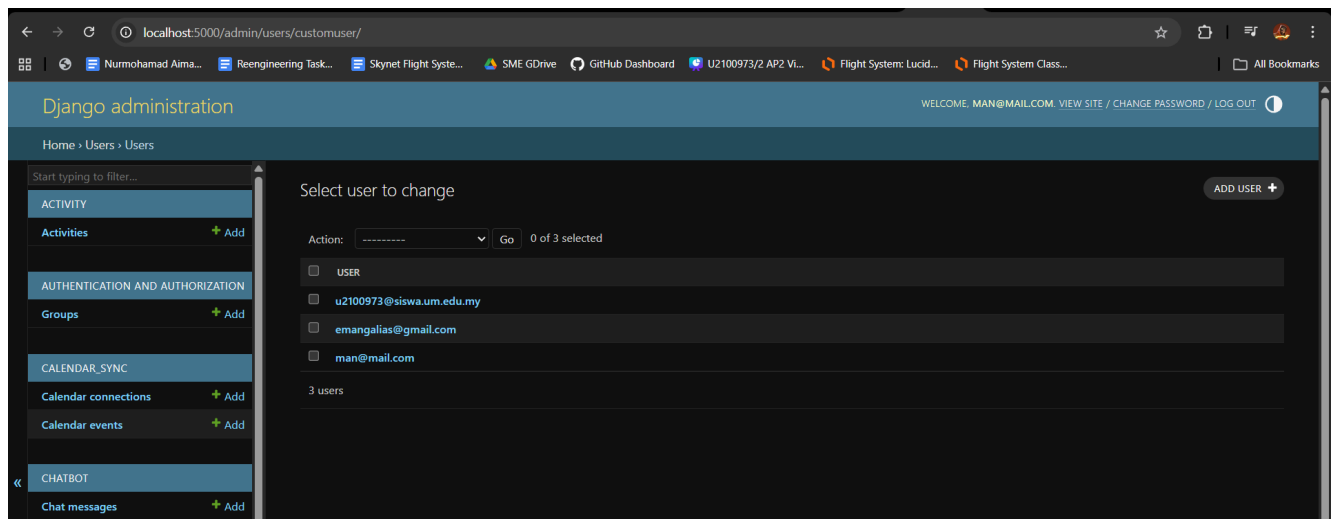


Figure 5.0: Admin Page

Screenshot shows the Django admin login page accessible at **`http://localhost:5000/admin`**, can also access the users database at **`http://localhost:5000/admin/users/customuser/`** confirming that the application is running correctly in the Docker container and is accessible via web browser.

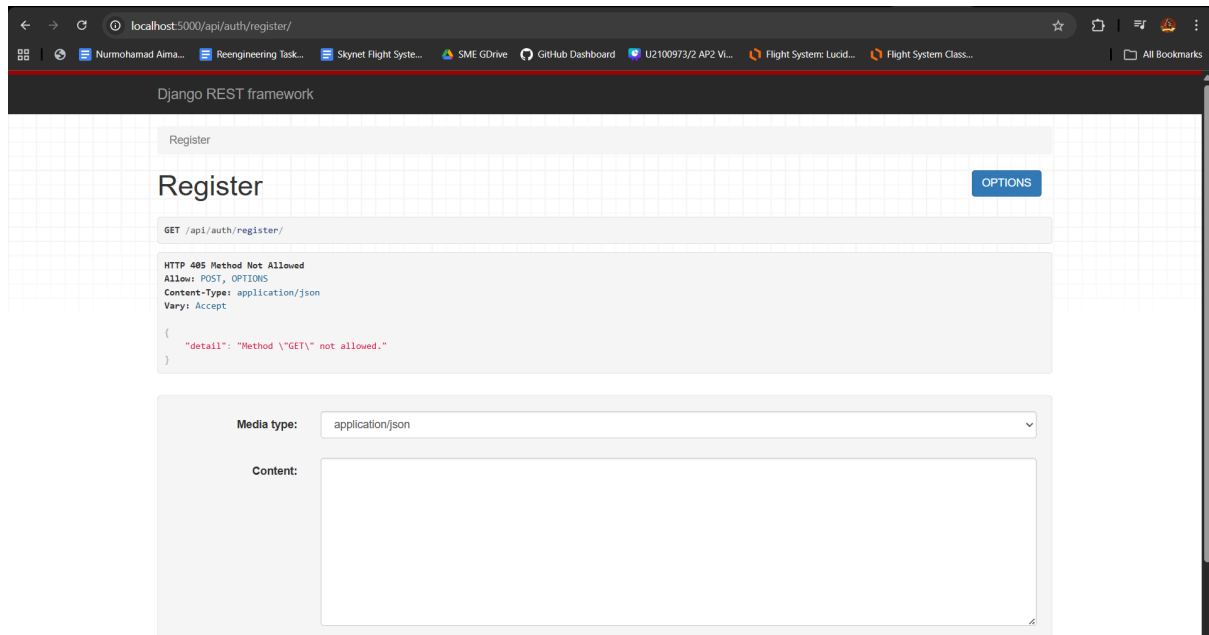


Figure 5.1: API Endpoint

Shows the Django REST framework browsable API interface at **<http://localhost:5000/api/auth/register/>**. The HTTP 405 Method Not Allowed is expected behavior (endpoint only accepts POST), confirming the API is properly configured and responding.

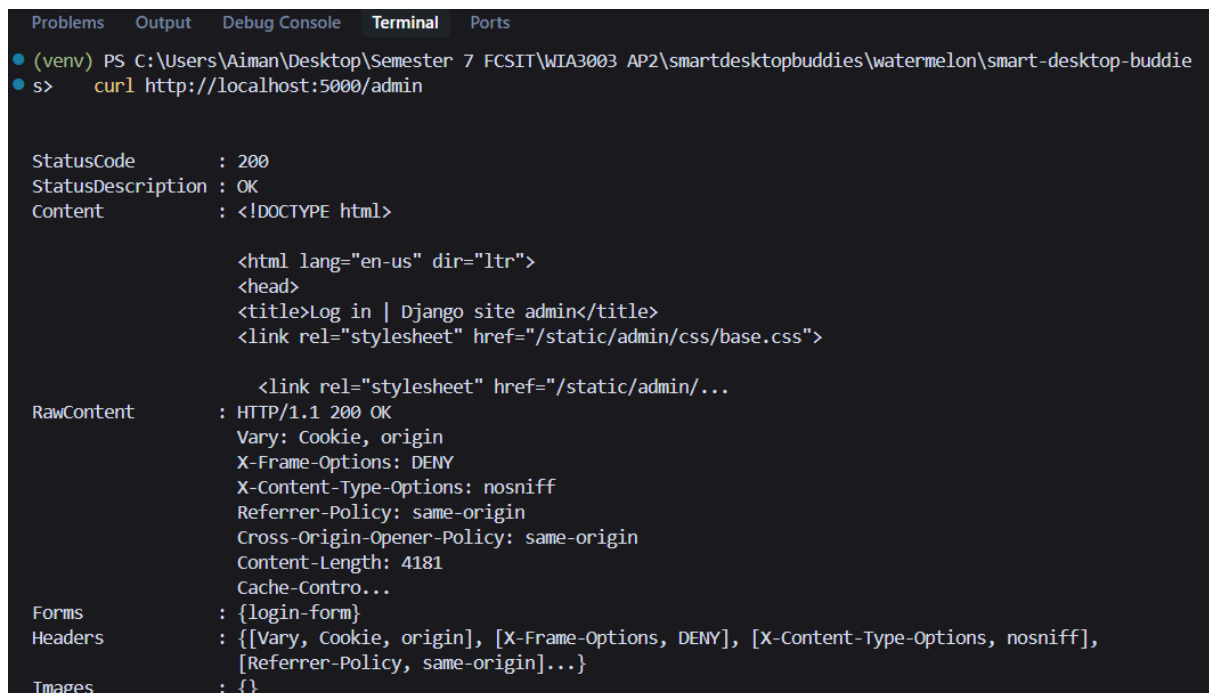


Figure 5.2: CURL Test (1/2)

```

Problems  Output  Debug Console  Terminal  Ports
Images    : {}
InputFields : {@{innerHTML=; innerText=; outerHTML=<INPUT type=hidden
value=v0QKZWQEGazAJHCJDvAKD9QI2v6arRvByN09NqI7ROsgXWRTxVE68pVldQfasDgh
name=csrfmiddlewaretoken>; outerText=; tagName=INPUT; type=hidden;
value=v0QKZWQEGazAJHCJDvAKD9QI2v6arRvByN09NqI7ROsgXWRTxVE68pVldQfasDgh;
name=csrfmiddlewaretoken}, @{{innerHTML=; innerText=; outerHTML=<INPUT id=id_username
value=v0QKZWQEGazAJHCJDvAKD9QI2v6arRvByN09NqI7ROsgXWRTxVE68pVldQfasDgh;
name=csrfmiddlewaretoken}, @{{innerHTML=; innerText=; outerHTML=<INPUT id=id_username
maxLength=254 name=username required autocomplete="username" autocapitalize="none"
autofocus>; outerText=; tagName=INPUT; id=id_username; maxLength=254; name=username;
required=; autocomplete=username; autocapitalize=none; autofocus=}, @{{innerHTML=;
innerText=; outerHTML=<INPUT id=id_password type=password value="" name=password required
autocomplete="current-password">; outerText=; tagName=INPUT; id=id_password; type=password;
value=; name=password; required=; autocomplete=current-password}, @{{innerHTML=; innerText=;
outerHTML=<INPUT type=hidden value=/admin/ name=next>; outerText=; tagName=INPUT;
type=hidden; value=/admin/; name=next}...}

Links      : {@{innerHTML=Skip to main content; innerText=Skip to main content; outerHTML=<A
class=skip-to-content-link href="#content-start">Skip to main content</A>; outerText=Skip
to main content; tagName=A; class=skip-to-content-link; href=#content-start},
@{{innerHTML=Django administration; innerText=Django administration; outerHTML=<A
href="/admin/">Django administration</A>; outerText=Django administration; tagName=A;
href=/admin/}}

Links      : {@{innerHTML=Skip to main content; innerText=Skip to main content; outerHTML=<A
class=skip-to-content-link href="#content-start">Skip to main content</A>; outerText=Skip
to main content; tagName=A; class=skip-to-content-link; href=#content-start},
@{{innerHTML=Django administration; innerText=Django administration; outerHTML=<A
href="/admin/">Django administration</A>; outerText=Django administration; tagName=A;
href=/admin/}}
to main content; tagName=A; class=skip-to-content-link; href=#content-start},
@{{innerHTML=Django administration; innerText=Django administration; outerHTML=<A
href="/admin/">Django administration</A>; outerText=Django administration; tagName=A;
href=/admin/}}
@{{innerHTML=Django administration; innerText=Django administration; outerHTML=<A
href="/admin/">Django administration</A>; outerText=Django administration; tagName=A;
href=/admin/}}
href="/admin/">Django administration</A>; outerText=Django administration; tagName=A;
href=/admin/}}
href="/admin/">Django administration</A>; outerText=Django administration; tagName=A;
href=/admin/}}

ParsedHtml : mshtml.HTMLDocumentClass
RawContentLength : 4181

```

Figure 5.3: CURL Test (2/2)

Screenshots show a **curl http://localhost:5000/admin** command testing the application endpoint, returning HTML content which confirms the server is responding correctly to HTTP requests.

```

(venv) PS C:\Users\Aiman\Desktop\Semester 7 FCSIT\WIA3003 AP2\smartdesktopbuddies\watermelon\smart-desktop-buddie
s> docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
32aa231b121f   smart-desktop-buddies              "/bin/sh -c 'python ..." 52 minutes ago Up 52 minutes 0.0.0.0:5000->8000/tcp, [::]:5000->8000/tcp
keen_torvalds

```

Figure 5.4: Running Container Verification

Screenshot shows the docker ps command output confirming the container is running with status "Up" and correct port mapping (0.0.0.0:5000->8000/tcp).

Conclusion

The Smart Desktop Buddies application was successfully containerized using Docker. The Dockerfile was created, the image was built, and the container runs with the application accessible at <http://localhost:5000>.