

# Report on Docker Assignment: urgentNews Application

## Introduction

This report outlines the creation and deployment of the Docker-based web application named "urgentNews." The task involves setting up Docker containers for a MySQL database, backend service, and frontend service to display the ten most recent urgent news articles.

## Application Overview

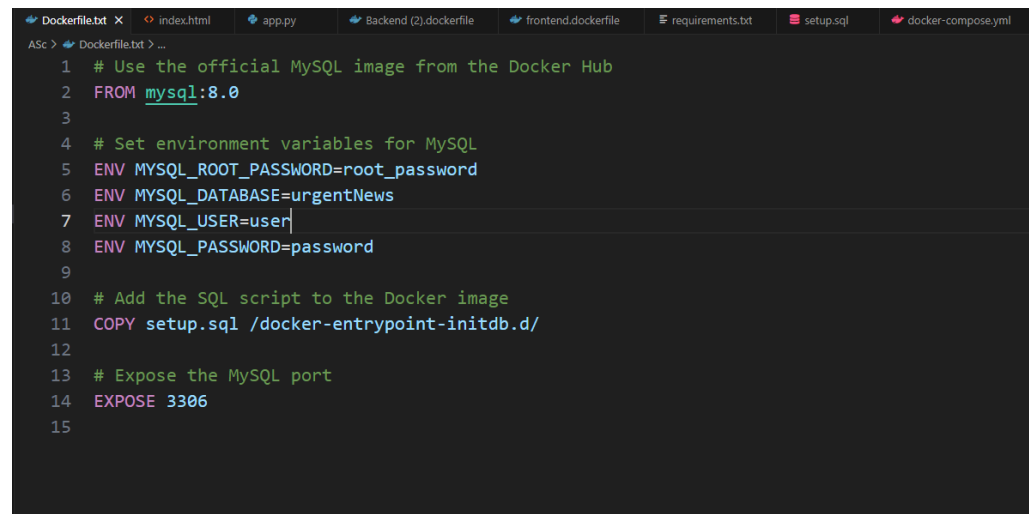
The urgentNews application is a web-based platform designed to display the ten most recent urgent news articles. The application runs on three Docker containers:

1. **Database Container:** Hosts a MySQL database, sets up a table named "News", and inserts ten articles.
2. **Backend Container:** Provides an API to retrieve news articles from the database using Flask.
3. **Frontend Container:** Hosts the frontend webpage using an Apache server.

## Docker Files and Explanations

### 1. Database Container

#### Dockerfile:

A screenshot of a code editor showing a Dockerfile for a MySQL database container. The file is named 'Dockerfile.txt' and is open in a window titled 'Dockerfile.txt'. The code is as follows:

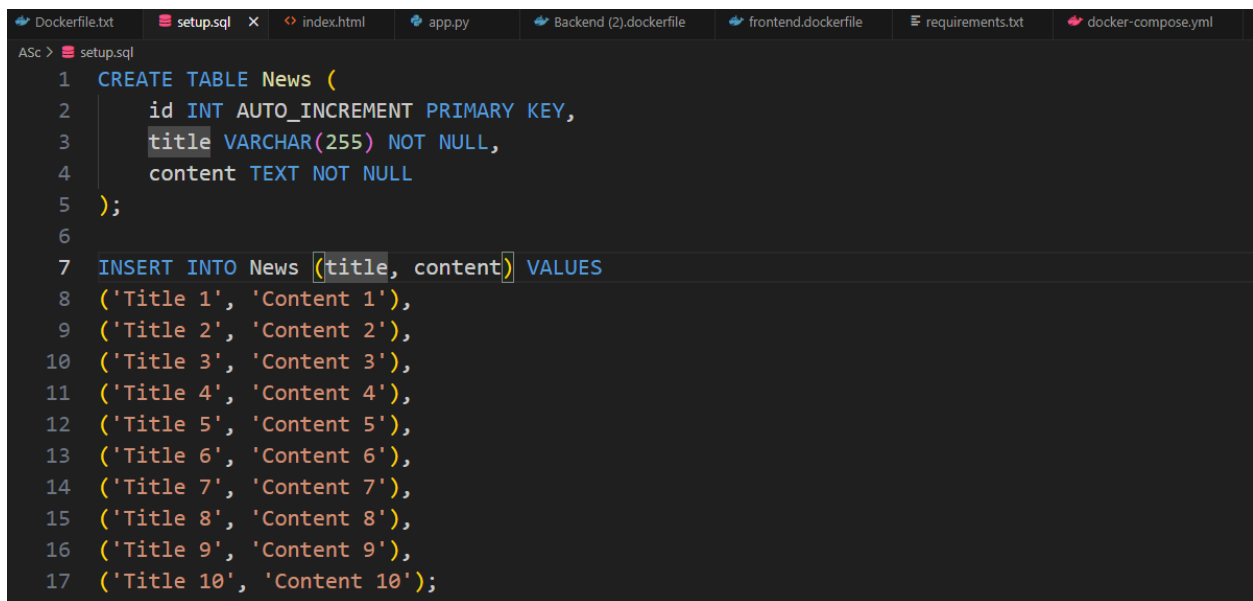
```
1 # Use the official MySQL image from the Docker Hub
2 FROM mysql:8.0
3
4 # Set environment variables for MySQL
5 ENV MYSQL_ROOT_PASSWORD=root_password
6 ENV MYSQL_DATABASE=urgentNews
7 ENV MYSQL_USER=user
8 ENV MYSQL_PASSWORD=password
9
10 # Add the SQL script to the Docker image
11 COPY setup.sql /docker-entrypoint-initdb.d/
12
13 # Expose the MySQL port
14 EXPOSE 3306
15
```

## Second semester 2023/2024 Cloud computing Homework Assignment # 2

### Rahaf sahir salman

#### Explanation:

- **FROM mysql:8.0:** Uses the official MySQL 8.0 image from Docker Hub.
- **\*\*ENV MYSQL\_\*:** Sets environment variables for MySQL, including the root password, database name, user, and password.
- **COPY setup.sql /docker-entrypoint-initdb.d/:** Copies the `setup.sql` file into the MySQL Docker image. This script initializes the database.
- **EXPOSE 3306:** Exposes port 3306 for MySQL.



```
1 CREATE TABLE News (
2     id INT AUTO_INCREMENT PRIMARY KEY,
3     title VARCHAR(255) NOT NULL,
4     content TEXT NOT NULL
5 );
6
7 INSERT INTO News (title, content) VALUES
8 ('Title 1', 'Content 1'),
9 ('Title 2', 'Content 2'),
10 ('Title 3', 'Content 3'),
11 ('Title 4', 'Content 4'),
12 ('Title 5', 'Content 5'),
13 ('Title 6', 'Content 6'),
14 ('Title 7', 'Content 7'),
15 ('Title 8', 'Content 8'),
16 ('Title 9', 'Content 9'),
17 ('Title 10', 'Content 10');
```

#### Explanation:

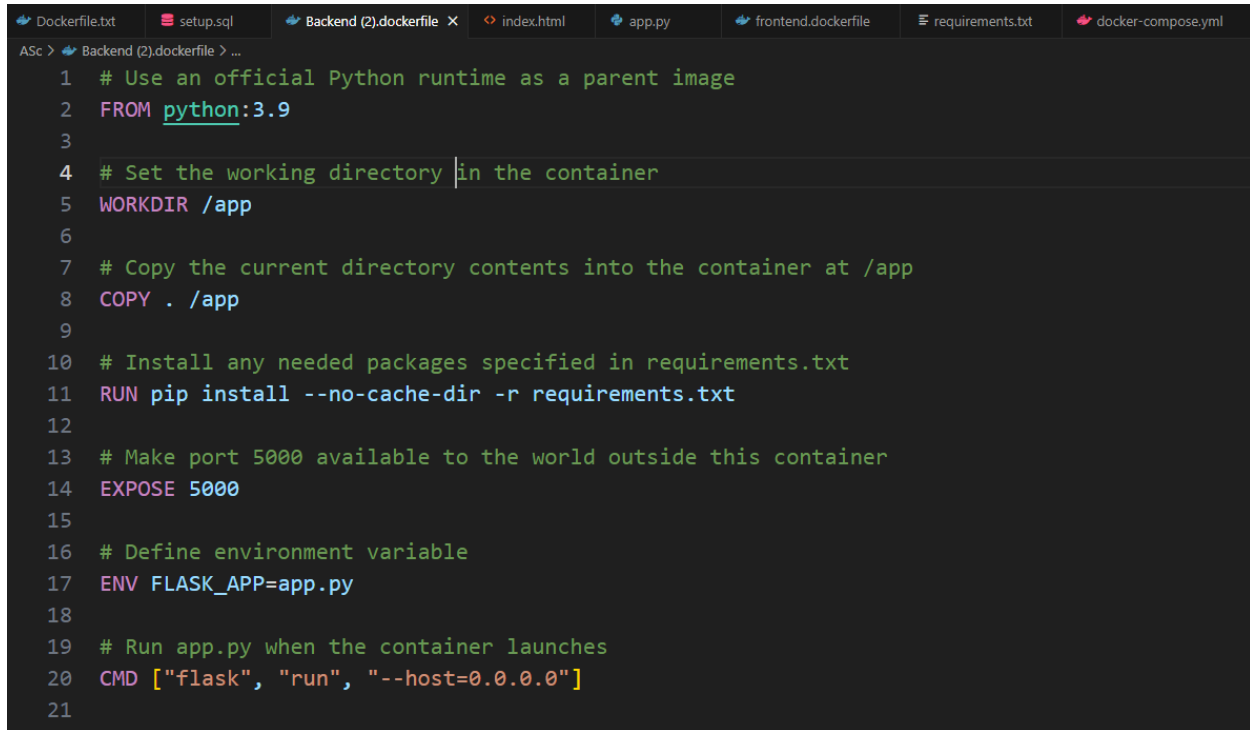
- This script creates a table named `News` with columns `id`, `title`, and `content`.
- It inserts ten rows of sample news data into the `News` table.

## Second semester 2023/2024 Cloud computing Homework Assignment # 2

### Rahaf sahir salman

#### 2. Backend Container

##### Dockerfile:



```
1 # Use an official Python runtime as a parent image
2 FROM python:3.9
3
4 # Set the working directory in the container
5 WORKDIR /app
6
7 # Copy the current directory contents into the container at /app
8 COPY . /app
9
10 # Install any needed packages specified in requirements.txt
11 RUN pip install --no-cache-dir -r requirements.txt
12
13 # Make port 5000 available to the world outside this container
14 EXPOSE 5000
15
16 # Define environment variable
17 ENV FLASK_APP=app.py
18
19 # Run app.py when the container launches
20 CMD ["flask", "run", "--host=0.0.0.0"]
21
```

##### Explanation:

- **FROM python:3.9:** Uses the official Python 3.9 image from Docker Hub.
- **WORKDIR /app:** Sets the working directory in the container to /app.
- **COPY . /app:** Copies the current directory contents into the container at /app.
- **RUN pip install --no-cache-dir -r requirements.txt:** Installs Python dependencies specified in requirements.txt.
- **EXPOSE 5000:** Exposes port 5000 for the Flask application.
- **ENV FLASK\_APP=app.py:** Sets an environment variable for the Flask app.
- **CMD ["flask", "run", "--host=0.0.0.0"]:** Runs the Flask application.

## Second semester 2023/2024 Cloud computing Homework Assignment # 2

### Rahaf sahir salman

app.py:

```
Dockerfile.txt  setup.sql  Backend (2).dockerfile  index.html  app.py  frontend.dockerfile  requirements.txt  docker-compose.yml
ASc > app.py > get_db_connection
1 from flask import Flask, jsonify
2 import mysql.connector
3
4 app = Flask(__name__)
5
6 def get_db_connection():
7     connection = mysql.connector.connect(
8         host='mysql-urgentnews',
9         user='root',
10        password='root_password',
11        database='urgentNews'
12    )
13    return connection
14
15 @app.route('/getUrgentNews', methods=['GET'])
16 def get_urgent_news():
17     connection = get_db_connection()
18     cursor = connection.cursor(dictionary=True)
19     cursor.execute('SELECT title, content FROM News')
20     news = cursor.fetchall()
21     cursor.close()
22     connection.close()
23     return jsonify(news)
```

Explanation:

- This Flask application defines an endpoint `/getUrgentNews` that retrieves news articles from the MySQL database and returns them as JSON.

requirements.txt:

```
Dockerfile.txt  setup.sql  Backend (2).dockerfile  index.html  app.py  frontend.dockerfile  requirements.txt  docker-compose.yml
ASc > requirements.txt
1 Flask
2 mysql-connector-python
3
```

Explanation:

- Lists the Python dependencies for the backend application, specifically Flask and mysql-connector-python.

## Second semester 2023/2024 Cloud computing Homework Assignment # 2

### Rahaf sahir salman

### 3. Frontend Container

#### Dockerfile:

```
Dockerfile.txt  setup.sql  Backend (2).dockerfile  index.html  app.py  frontend.dockerfile X  requirements.txt  docker-compose.yml
ASc > frontend.dockerfile > ...
1 FROM httpd:latest
2
3 COPY ./public-html/ /usr/local/apache2/htdocs/
4
5 EXPOSE 80
6
```

#### Explanation:

- **FROM httpd**  
: Uses the official Apache HTTP server image from Docker Hub.
- **COPY ./public-html/ /usr/local/apache2/htdocs/**: Copies the contents of the `public-html` directory to the Apache server's document root.
- **EXPOSE 80**: Exposes port 80 for the Apache server.

#### index.html (located in public-html directory):

```
Dockerfile.txt  setup.sql  Backend (2).dockerfile  index.html X  app.py  frontend.dockerfile  requirements.txt  docker-compose.yml
ASc > public-html > index.html > html > body > div#news
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Urgent News</title>
8 </head>
9
10 <body>
11   <h1>Urgent News</h1>
12   <div id="news"></div>
13   <script>
14     fetch('http://localhost:5000/getUrgentNews') // والمنفذ الصحيح localhost تم تعديل هنا لاستخدام
15     .then(response => response.json())
16     .then(data => {
17       const newsDiv = document.getElementById('news');
18       data.forEach(article => {
19         const articleDiv = document.createElement('div');
20         const title = document.createElement('h2');
21         const content = document.createElement('p');
22         title.textContent = article.title;
23         content.textContent = article.content;
24       });
25     });
26   </script>
27
```

## Second semester 2023/2024 Cloud computing Homework Assignment # 2

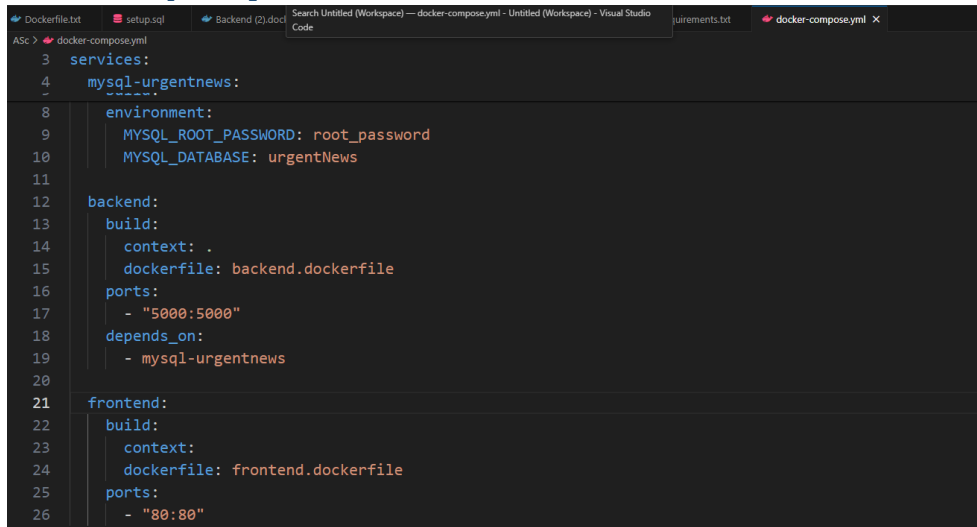
Rahaf sahir salman

### Explanation:

- This HTML file defines the structure of the webpage. It includes a script that fetches news data from the backend API and displays it on the page.

### Docker Compose File

docker-compose.yml



```
3  services:
4    mysql-urgentnews:
5      image: mysql:8.0
6      environment:
7        MYSQL_ROOT_PASSWORD: root_password
8        MYSQL_DATABASE: urgentNews
9
10   backend:
11     build:
12       context: .
13       dockerfile: backend.dockerfile
14     ports:
15       - "5000:5000"
16     depends_on:
17       - mysql-urgentnews
18
19   frontend:
20     build:
21       context: .
22       dockerfile: frontend.dockerfile
23     ports:
24       - "80:80"
```

### Explanation:

- **version: '3'**: Specifies the version of Docker Compose.
- **services**: Defines the services (containers) required for the application.

#### mysql-urgentnews:

- **build**: Specifies the build context and Dockerfile for the MySQL container.
- **environment**: Sets environment variables for MySQL.

#### backend:

- **build**: Specifies the build context and Dockerfile for the backend container.
- **ports**: Maps port 5000 of the container to port 5000 on the host.
- **depends\_on**: Ensures the backend service starts after the MySQL container.

## Second semester 2023/2024 Cloud computing Homework Assignment # 2

Rahaf sahir salman

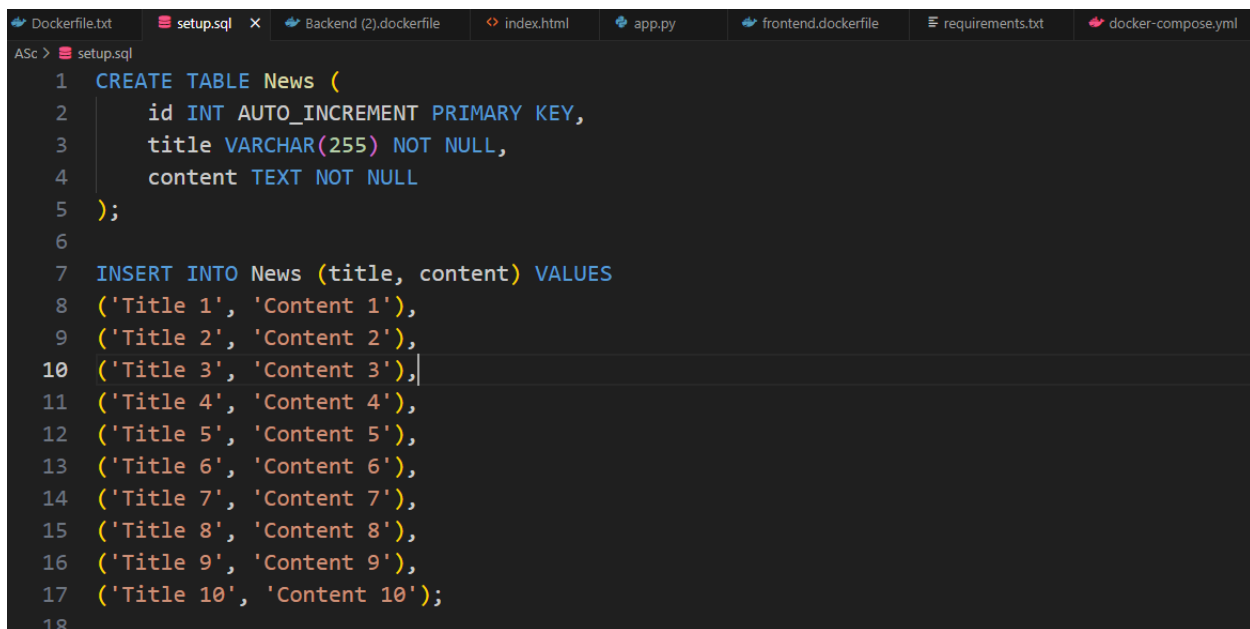
frontend:

- **build:** Specifies the build context and Dockerfile for the frontend container.
- **ports:** Maps port 80 of the container to port 80 on the host.

Explanation:

- **FROM mysql:8.0:** Uses the official MySQL 8.0 image from Docker Hub.
- **\*\*ENV MYSQL\_\*:** Sets environment variables for MySQL, including the root password, database name, user, and password.
- **COPY setup.sql /docker-entrypoint-initdb.d/:** Copies the `setup.sql` file into the MySQL Docker image. This script initializes the database.
- **EXPOSE 3306:** Exposes port 3306 for MySQL.

setup.sql:



```
1 CREATE TABLE News (
2     id INT AUTO_INCREMENT PRIMARY KEY,
3     title VARCHAR(255) NOT NULL,
4     content TEXT NOT NULL
5 );
6
7 INSERT INTO News (title, content) VALUES
8 ('Title 1', 'Content 1'),
9 ('Title 2', 'Content 2'),
10 ('Title 3', 'Content 3'),
11 ('Title 4', 'Content 4'),
12 ('Title 5', 'Content 5'),
13 ('Title 6', 'Content 6'),
14 ('Title 7', 'Content 7'),
15 ('Title 8', 'Content 8'),
16 ('Title 9', 'Content 9'),
17 ('Title 10', 'Content 10');
```

Explanation:

- This script creates a table named `News` with columns `id`, `title`, and `content`.
- It inserts ten rows of sample news data into the `News` table.

Explanation:

## Second semester 2023/2024 Cloud computing Homework Assignment # 2

Rahaf sahir salman

•

### Detailed Instructions to Run the Application

1. **Create the necessary Dockerfiles and file structure:**
  - Create the `docker-compose.yml` file.
  - Create `mysql.dockerfile` for the MySQL container.
  - Create `backend.dockerfile` for the backend container.
  - Create `frontend.dockerfile` for the frontend container.
  - Create `setup.sql` file.
  - Create `app.py` and `requirements.txt` for the backend container.
  - Create the `public-html` directory and place the `index.html` file inside it.
2. **Build and start the containers using Docker Compose:**

```
docker-compose up --build
```

- This command builds the Docker images and starts the containers as defined in the `docker-compose.yml` file.
3. **Access the Application:**
    - Open a web browser and navigate to `http://localhost`. The webpage should display the ten most recent urgent news articles.

### Detailed Instructions to Run the Application

- **Pull the Docker images from DockerHub:**

```
docker pull dockerhubusername/urgentnews-db
docker pull dockerhubusername/urgentnews-backend
docker pull dockerhubusername/urgentnews-frontend
```

- **Run the MySQL database container:**

```
docker run -d --name db -e MYSQL_ROOT_PASSWORD=root_password -e
MYSQL_DATABASE=urgentNews -e MYSQL_USER=user -e MYSQL_PASSWORD=password
dockerhubusername/urgentnews-db
```

•



## Second semester 2023/2024 Cloud computing Homework Assignment # 2

**Rahaf sahir salman**

- 
- 
- 

This command runs the MySQL database container with the environment variables set for root password, database name, user, and password.

- **Run the backend service container:**

```
docker run -d --name backend --link db:mysql-urgentnews -p 5000:5000  
dockerhubusername/urgentnews-backend
```

- This command runs the backend service container, linking it to the database container (db) and exposing port 5000.

- **Run the frontend service container:**

```
docker run -d --name frontend -p 80:80 dockerhubusername/urgentnews-frontend
```

- This command runs the frontend service container and exposes port 80.

- **Access the Application:**

- Open a web browser and navigate to <http://localhost>. The webpage should display the ten most recent urgent news articles.

### [Docker Hub Repository](#)

The Docker images for this application are hosted on Docker Hub and can be accessed at

**<https://hub.docker.com/u/rahafsalman7>**

### [Conclusion](#)

This report provides a detailed overview of the urgentNews application, explaining the Dockerfile configurations, database setup, backend service implementation, frontend setup, and the steps to deploy the application. The urgentNews application demonstrates a successful use of Docker containers to create a full-stack web application.