در این بخش، یک پروژه داکر برای حل معادله درجه دوم با استفاده از **پایتون** در سیستم‌عامل **اوبونتو** را ایجاد می‌کنیم. در این پروژه، همه فایل‌ها و نام‌ها به‌عنوان "sadra" در نظر گرفته می‌شوند.

**مراحل ایجاد پروژه داکر برای حل معادله درجه دوم**

**گام 1: ایجاد پوشه پروژه**

ابتدا پوشه‌ای به نام **sadra** ایجاد کنید:

bash

mkdir sadra

cd sadra

**گام 2: ایجاد فایل Python**

یک فایل پایتون به نام **sadra.py** ایجاد کنید که شامل کد حل معادله درجه دوم باشد:

bash

nano sadra.py

سپس کد زیر را در این فایل وارد کنید:

python

import cmath

def solve\_quadratic(a, b, c):

# محاسبه دلتا

delta = b\*\*2 - 4\*a\*c

# محاسبه ریشه‌ها

root1 = (-b + cmath.sqrt(delta)) / (2 \* a)

root2 = (-b - cmath.sqrt(delta)) / (2 \* a)

return root1, root2

if \_\_name\_\_ == "\_\_main\_\_":

# ورودی از کاربر

a = float(input("Enter coefficient a: "))

b = float(input("Enter coefficient b: "))

c = float(input("Enter coefficient c: "))

# حل معادله

roots = solve\_quadratic(a, b, c)

# نمایش ریشه‌ها

print(f"The roots of the equation are: {roots[0]} and {roots[1]}")

این کد یک معادله درجه دوم را حل کرده و ریشه‌ها را نمایش می‌دهد.

**گام 3: ایجاد فایل Dockerfile**

یک فایل به نام **Dockerfile** ایجاد کنید:

bash

nano Dockerfile

محتوای زیر را در فایل **Dockerfile** وارد کنید:

Dockerfile

# استفاده از تصویر اوبونتو

FROM ubuntu:20.04

# نصب پایتون و pip

RUN apt-get update && apt-get install -y python3 python3-pip

# تعیین دایرکتوری کاری

WORKDIR /app

# کپی فایل پایتون به داخل کانتینر

COPY sadra.py .

# اجرای برنامه پایتون

CMD ["python3", "sadra.py"]

این فایل دستورالعمل‌هایی را برای داکر فراهم می‌کند تا یک کانتینر بر اساس اوبونتو ایجاد کند و برنامه پایتون شما را در آن اجرا کند.

**گام 4: ساخت تصویر داکر**

برای ساخت تصویر داکر، دستور زیر را در ترمینال اجرا کنید:

bash

docker build -t sadra-project .

این دستور یک تصویر داکر به نام **sadra-project** می‌سازد که برنامه پایتون را برای حل معادله درجه دوم در بر دارد.

**گام 5: اجرای کانتینر**

پس از ساخت تصویر، کانتینر را با دستور زیر اجرا کنید:

bash

docker run -it --name sadra-container sadra-project

استفاده از گزینه -it این امکان را می‌دهد که با کانتینر به‌صورت تعاملی (interactive) کار کنید و ورودی‌ها را از کاربر بگیرید.

**گام 6: اجرای برنامه و مشاهده خروجی**

وقتی کانتینر اجرا شد، برنامه پایتون از شما درخواست می‌کند تا ضرایب aaa، bbb و ccc را وارد کنید. به‌عنوان مثال، فرض کنید معادله x2+5x+6=0x^2 + 5x + 6 = 0x2+5x+6=0 را حل می‌کنید:

mathematica

Enter coefficient a: 1

Enter coefficient b: 5

Enter coefficient c: 6

برنامه ریشه‌های معادله را محاسبه کرده و نتیجه زیر را به شما نمایش می‌دهد:

sql

The roots of the equation are: (-2+0j) and (-3+0j)

**گام 7: توقف و حذف کانتینر (اختیاری)**

اگر بخواهید کانتینر را متوقف کرده یا حذف کنید، از دستورات زیر استفاده کنید:

bash

docker stop sadra-container

docker rm sadra-container

همچنین می‌توانید تصویر را نیز حذف کنید:

bash

docker rmi sadra-project

**مراحل ایجاد پروژه داکر برای حل معادله درجه دوم در پورت 8080**

**گام 1: ایجاد پوشه پروژه**

ابتدا پوشه‌ای به نام **sadra2** ایجاد کنید:

bash

mkdir sadra2

cd sadra2

**گام 2: ایجاد فایل Python با استفاده از Flask**

یک فایل پایتون به نام **sadra2.py** ایجاد کنید که با استفاده از Flask یک برنامه وب ساده ایجاد می‌کند:

bash

nano app.py

سپس کد زیر را در این فایل وارد کنید:

python

from flask import Flask, request, render\_template

import cmath

app = Flask(\_\_name\_\_)

# صفحه اصلی

@app.route('/')

def home():

return render\_template('index.html')

# حل معادله درجه دوم

@app.route('/solve', methods=['POST'])

def solve():

try:

a = float(request.form['a'])

b = float(request.form['b'])

c = float(request.form['c'])

delta = b\*\*2 - 4\*a\*c

root1 = (-b + cmath.sqrt(delta)) / (2 \* a)

root2 = (-b - cmath.sqrt(delta)) / (2 \* a)

return render\_template('result.html', root1=root1, root2=root2)

except ValueError:

return "Invalid input. Please enter valid numbers."

if \_\_name\_\_ == '\_\_main\_\_':

app.run(host='0.0.0.0', port=8080)

در این کد، یک سرور Flask ساده راه‌اندازی می‌شود که در پورت 8080 اجرا شده و معادله درجه دوم را حل می‌کند.

**گام 3: ایجاد قالب‌های HTML**

**ایجاد فایل index.html:**

ابتدا فایل HTML برای صفحه اصلی (فرم دریافت ورودی) را ایجاد کنید:

bash

mkdir templates

nano templates/index.html

سپس کد زیر را وارد کنید:

html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Quadratic Equation Solver</title>

</head>

<body>

<h1>Solve Quadratic Equation</h1>

<form action="/solve" method="POST">

<label for="a">Coefficient a:</label>

<input type="text" id="a" name="a" required><br><br>

<label for="b">Coefficient b:</label>

<input type="text" id="b" name="b" required><br><br>

<label for="c">Coefficient c:</label>

<input type="text" id="c" name="c" required><br><br>

<button type="submit">Solve</button>

</form>

</body>

</html>

این فرم سه ورودی برای ضرایب aaa، bbb، و ccc فراهم می‌کند.

**ایجاد فایل result.html:**

یک فایل دیگر برای نمایش نتایج به نام **result.html** ایجاد کنید:

bash

nano templates/result.html

سپس کد زیر را وارد کنید:

html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Result</title>

</head>

<body>

<h1>Results</h1>

<p>Root 1: {{ root1 }}</p>

<p>Root 2: {{ root2 }}</p>

<a href="/">Solve another equation</a>

</body>

</html>

این فایل نتایج معادله را نمایش می‌دهد و گزینه‌ای برای بازگشت به صفحه اصلی دارد.

**گام 4: ایجاد Dockerfile**

یک فایل به نام **Dockerfile** ایجاد کنید:

bash

nano Dockerfile

محتوای زیر را در فایل **Dockerfile** وارد کنید:

Dockerfile

# استفاده از تصویر پایه اوبونتو

FROM ubuntu:20.04

# نصب پایتون و ابزارهای مورد نیاز

RUN apt-get update && apt-get install -y python3 python3-pip

# نصب Flask

RUN pip3 install Flask

# تعیین دایرکتوری کاری

WORKDIR /app

# کپی فایل‌های پروژه به کانتینر

COPY . /app

# اجرای برنامه پایتون

CMD ["python3", "sadra2.py"]

این Dockerfile شامل مراحل نصب اوبونتو، پایتون، Flask و کپی کردن فایل‌های پروژه به داخل کانتینر است.

**گام 5: ساخت تصویر داکر**

برای ساخت تصویر داکر، دستور زیر را در ترمینال اجرا کنید:

bash

docker build -t sadra2 .

این دستور تصویر داکر را با نام **sadra2-project** ایجاد می‌کند.

**گام 6: اجرای کانتینر**

برای اجرای کانتینر در پورت **8080**، از دستور زیر استفاده کنید:

bash

docker run -d -p 8080:8080 --name sadra2-container sadra2

این دستور کانتینر را در پس‌زمینه اجرا می‌کند و پورت **8080** کانتینر را به پورت **8080** ماشین محلی شما متصل می‌کند.

**گام 7: دسترسی به برنامه در مرورگر**

اکنون می‌توانید به مرورگر خود رفته و به آدرس زیر بروید:

arduino

http://localhost:8080

در اینجا یک فرم برای وارد کردن ضرایب معادله درجه دوم مشاهده خواهید کرد. پس از وارد کردن ضرایب و کلیک بر روی "Solve"، ریشه‌های معادله نمایش داده خواهند شد.

**مثال:**

* اگر ضرایب زیر را وارد کنید:
  + a=1a = 1a=1
  + b=5b = 5b=5
  + c=6c = 6c=6

نتیجه باید به صورت زیر نمایش داده شود:

mathematica

Root 1: (-2+0j)

Root 2: (-3+0j)

**گام 8: توقف و حذف کانتینر (اختیاری)**

برای توقف و حذف کانتینر، از دستورات زیر استفاده کنید:

bash

docker stop sadra2-container

docker rm sadra2-container

همچنین می‌توانید تصویر را حذف کنید:

bash

docker rmi sadra2-project

docker network ls

docker stop <container\_name>

docker network rm <network\_name>

**جمع‌بندی**

در این پروژه، با استفاده از Flask و داکر، یک برنامه وب ساده برای حل معادله درجه دوم ایجاد کردیم که پارامترها را از طریق مرورگر دریافت کرده و نتایج را بر روی پورت **8080** نمایش می‌دهد. این پروژه به شما کمک می‌کند تا برنامه‌های وب را در محیط‌های ایزوله داکر اجرا و مدیریت کنید.