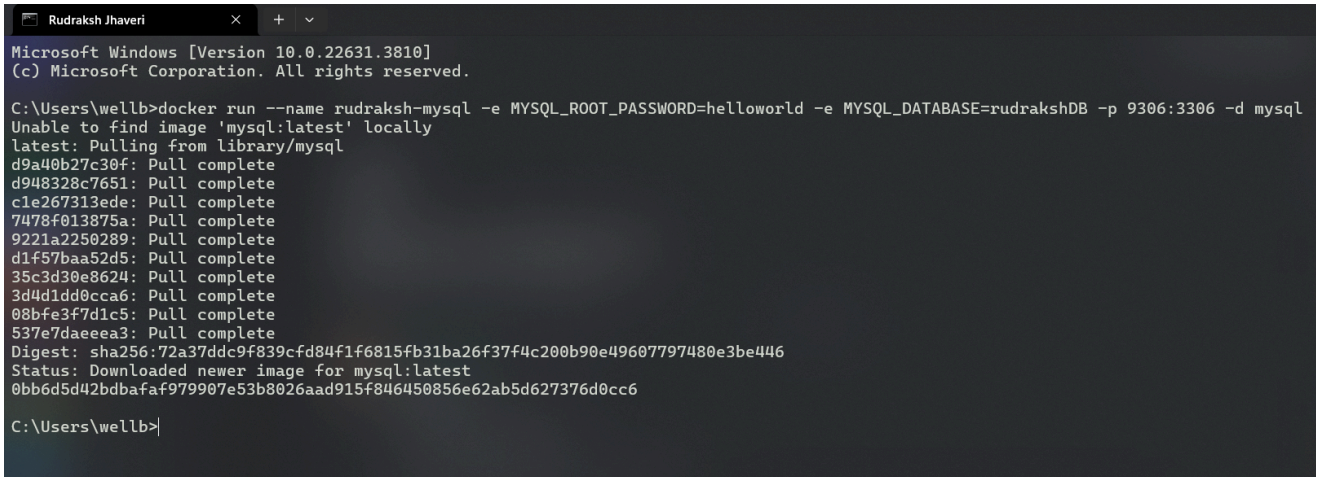
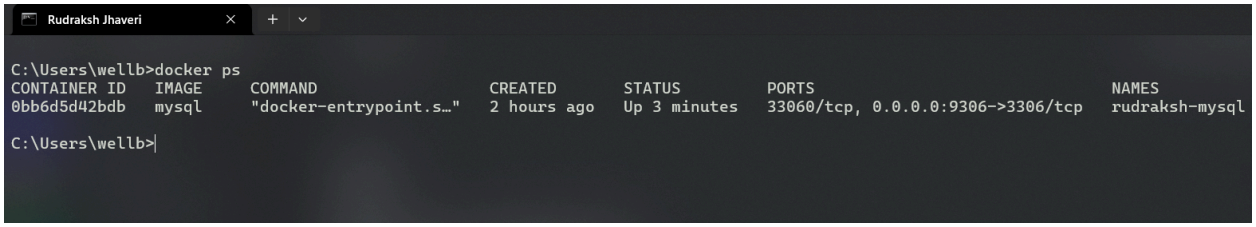
Task 11

Setting Up MySQL with Docker and Vanilla JavaScript for Testing Database Connections

1. Set up MySQL Using Docker



2. Verify Container Status



3. Test Database Connection Using Vanilla JavaScript.

- **npm init -y**

```
D:\Rudraksh_mysql_test>npm init -y
Wrote to D:\Rudraksh_mysql_test\package.json:

{
  "name": "rudraksh_mysql_test",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "mysql2": "^3.10.2"
  },
  "devDependencies": {},
  "description": ""
}
```

- **npm i mysql2**

```
D:\Rudraksh_mysql_test>npm i mysql2

up to date, audited 13 packages in 1s

found 0 vulnerabilities
```

- **Create an `index.js` file and write the code in it.**

index.js

```
const mysql = require('mysql2');

const connection = mysql.createConnection({
    host: 'localhost',
    port: '9306',
    user: 'root',
    password: 'helloworld',
    database: 'rudrakshDB'
});

connection.connect((err) => {
    if (err) {
        console.error('Error connecting to MySQL:', err.stack);
        return;
    }
    console.log('Connected to MySQL as ID:', connection.threadId);


    connection.end();
});

```

- **node index.js**

```
D:\Rudraksh_mysql_test>node index.js
Connected to MySQL as ID: 9

D:\Rudraksh_mysql_test>
```