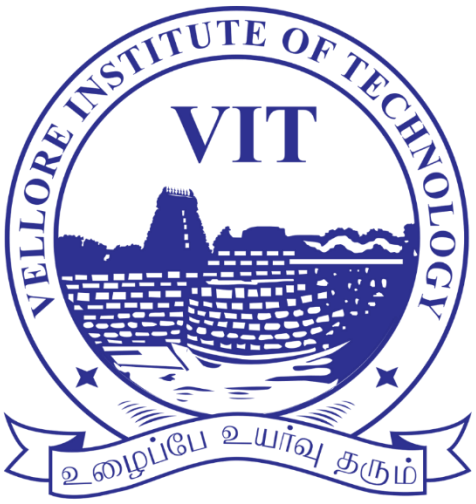


CSI3023

Advanced Server -Side Programming

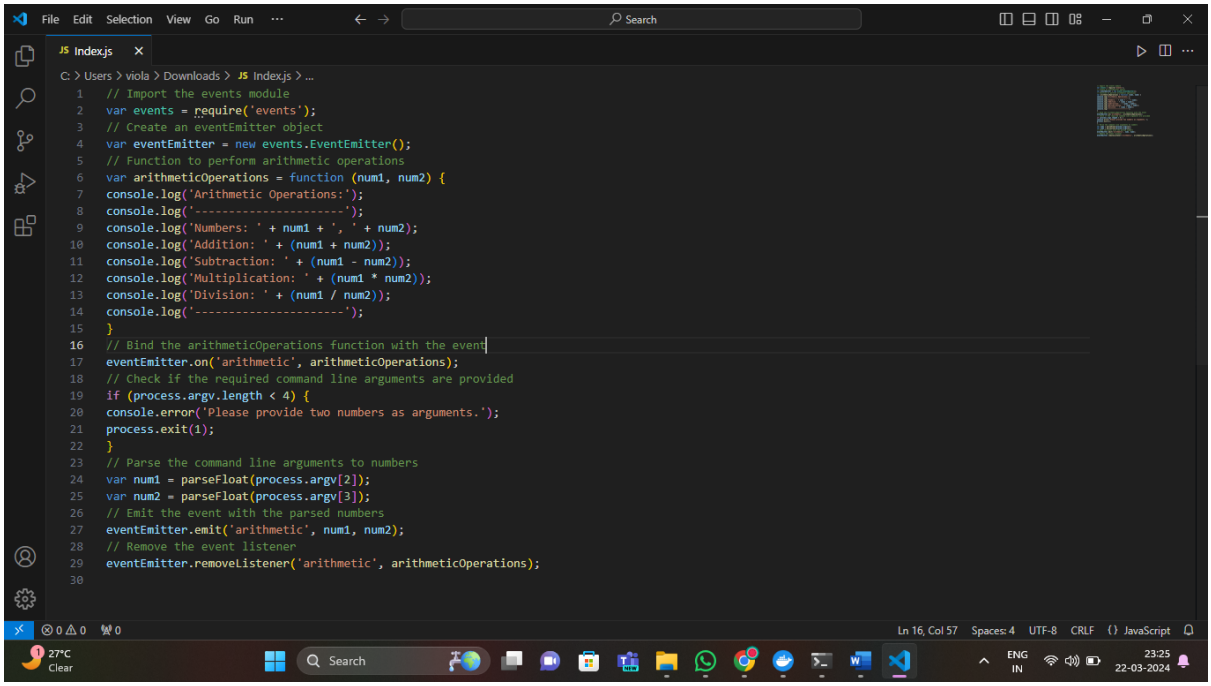
Winter Semester 2023 – '24

Digital Assignment - 6

SARAGA S
20MIC0081

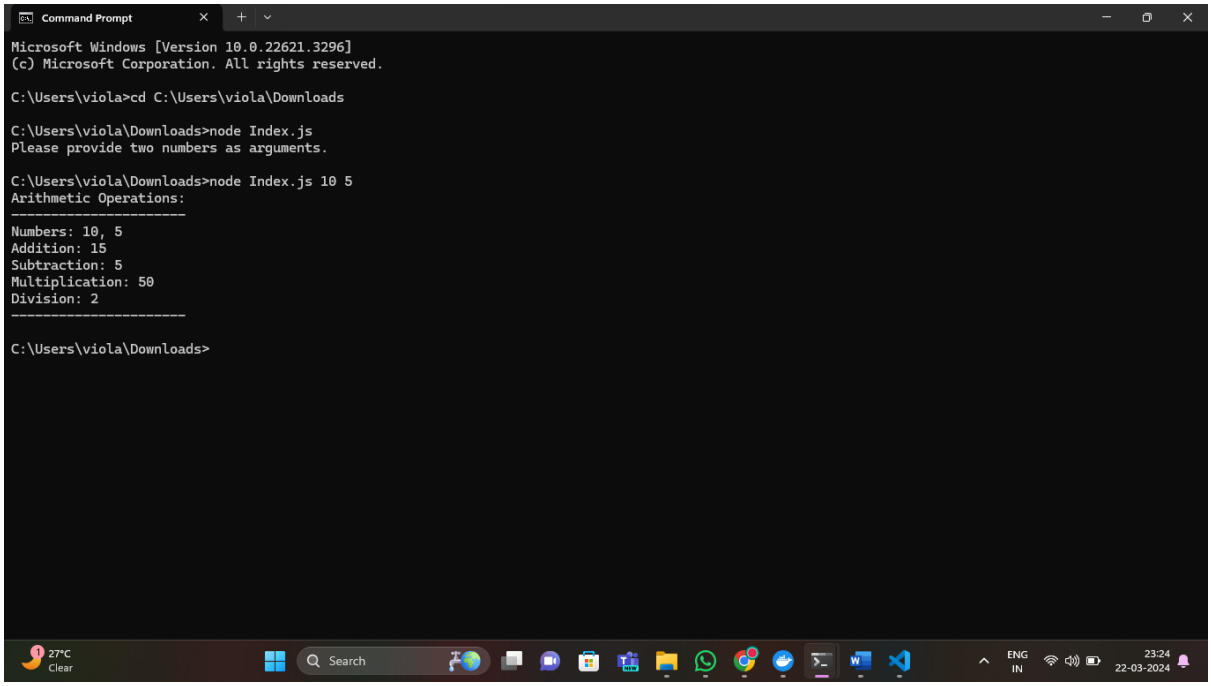
CODE:

```
// Import the events module
var events = require('events');
// Create an eventEmitter object
var eventEmitter = new events.EventEmitter();
// Function to perform arithmetic operations
var arithmeticOperations = function (num1, num2) {
console.log('Arithmetic Operations:');
console.log('----------------------');
console.log('Numbers: ' + num1 + ', ' + num2);
console.log('Addition: ' + (num1 + num2));
console.log('Subtraction: ' + (num1 - num2));
console.log('Multiplication: ' + (num1 * num2));
console.log('Division: ' + (num1 / num2));
console.log('----------------------');
}
// Bind the arithmeticOperations function with the event
eventEmitter.on('arithmetic', arithmeticOperations);
// Check if the required command line arguments are provided
if (process.argv.length < 4) {
console.error('Please provide two numbers as arguments.');
process.exit(1);
}
// Parse the command line arguments to numbers
var num1 = parseFloat(process.argv[2]);
var num2 = parseFloat(process.argv[3]);
// Emit the event with the parsed numbers
eventEmitter.emit('arithmetic', num1, num2);
// Remove the event listener
eventEmitter.removeListener('arithmetic', arithmeticOperations);
```



```
// Import the events module
var events = require('events');
// Create an eventEmitter object
var eventEmitter = new events.EventEmitter();
// Function to perform arithmetic operations
var arithmeticOperations = function (num1, num2) {
console.log('Arithmetic Operations:');
console.log('----------------------');
console.log('Numbers: ' + num1 + ', ' + num2);
console.log('Addition: ' + (num1 + num2));
console.log('Subtraction: ' + (num1 - num2));
console.log('Multiplication: ' + (num1 * num2));
console.log('Division: ' + (num1 / num2));
console.log('----------------------');
}
// Bind the arithmeticOperations function with the event
eventEmitter.on('arithmetic', arithmeticOperations);
// Check if the required command line arguments are provided
if (process.argv.length < 4) {
console.error('Please provide two numbers as arguments.');
process.exit(1);
}
// Parse the command line arguments to numbers
var num1 = parseFloat(process.argv[2]);
var num2 = parseFloat(process.argv[3]);
// Emit the event with the parsed numbers
eventEmitter.emit('arithmetic', num1, num2);
// Remove the event listener
eventEmitter.removeListener('arithmetic', arithmeticOperations);
```

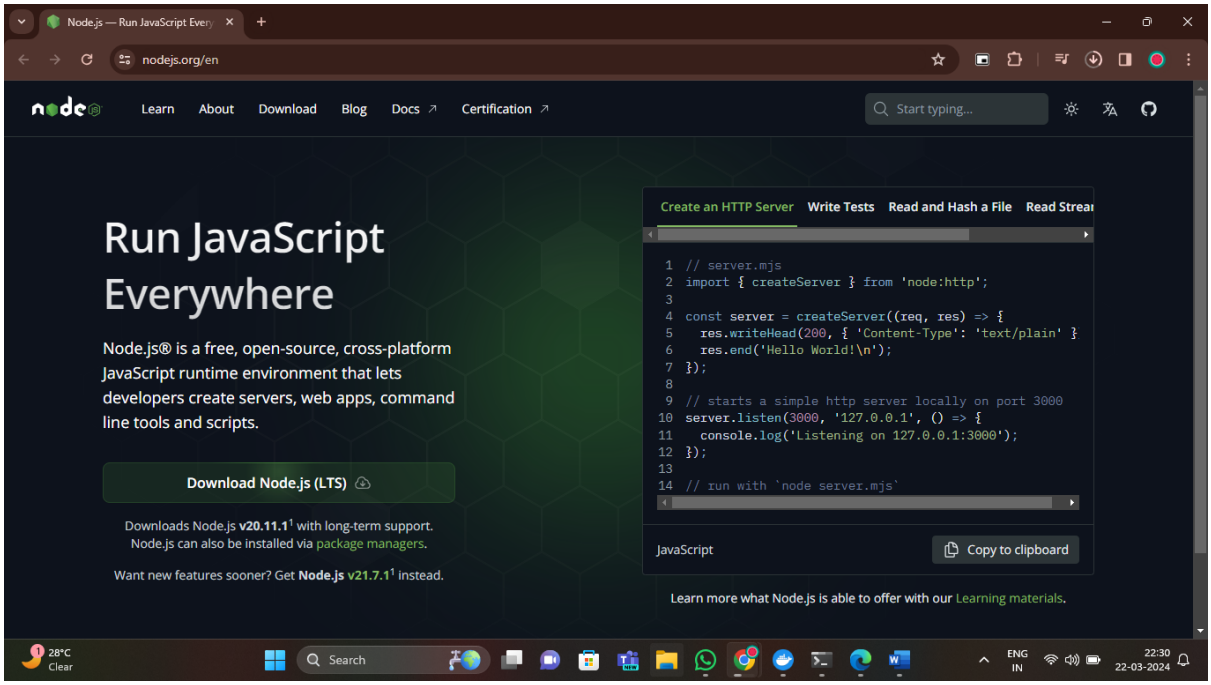
Output:



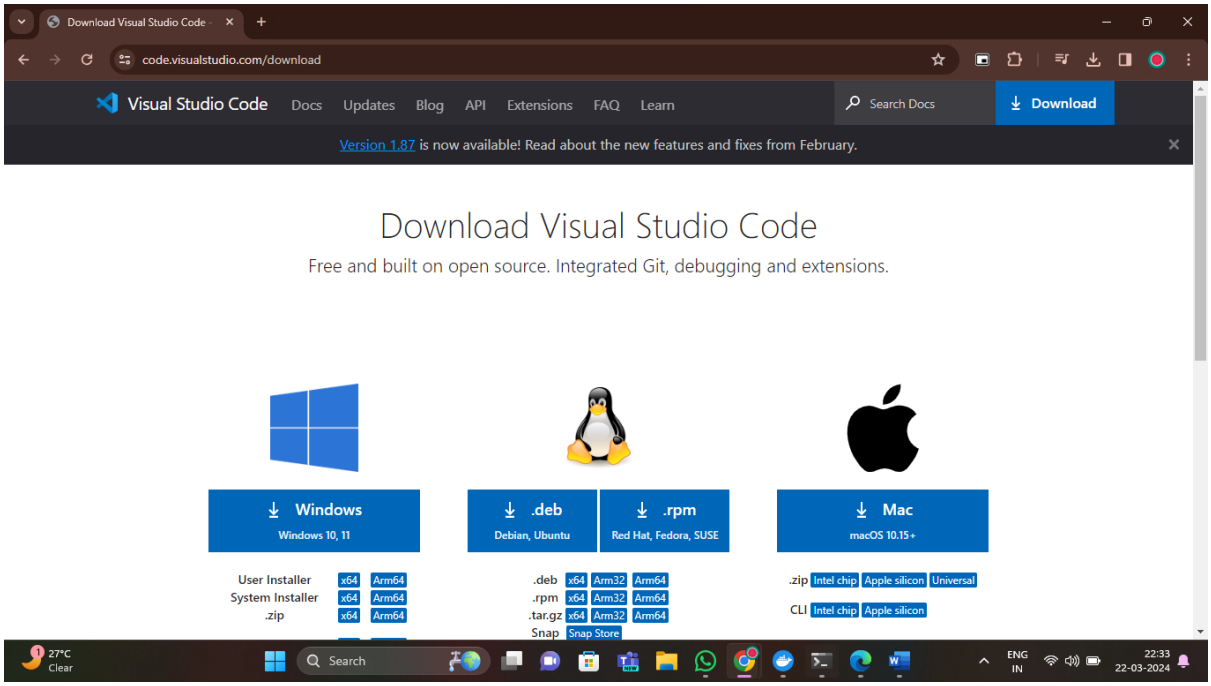
```
Microsoft Windows [Version 10.0.22621.3296]
(c) Microsoft Corporation. All rights reserved.

C:\Users\viola>cd C:\Users\viola\Downloads

C:\Users\viola\Downloads>node Index.js
Please provide two numbers as arguments.

C:\Users\viola\Downloads>node Index.js 10 5
Arithmetic Operations:
----------------------
Numbers: 10, 5
Addition: 15
Subtraction: 5
Multiplication: 50
Division: 2
----------------------

C:\Users\viola\Downloads>
```

Create and Deploy a Web application on Azure platform.

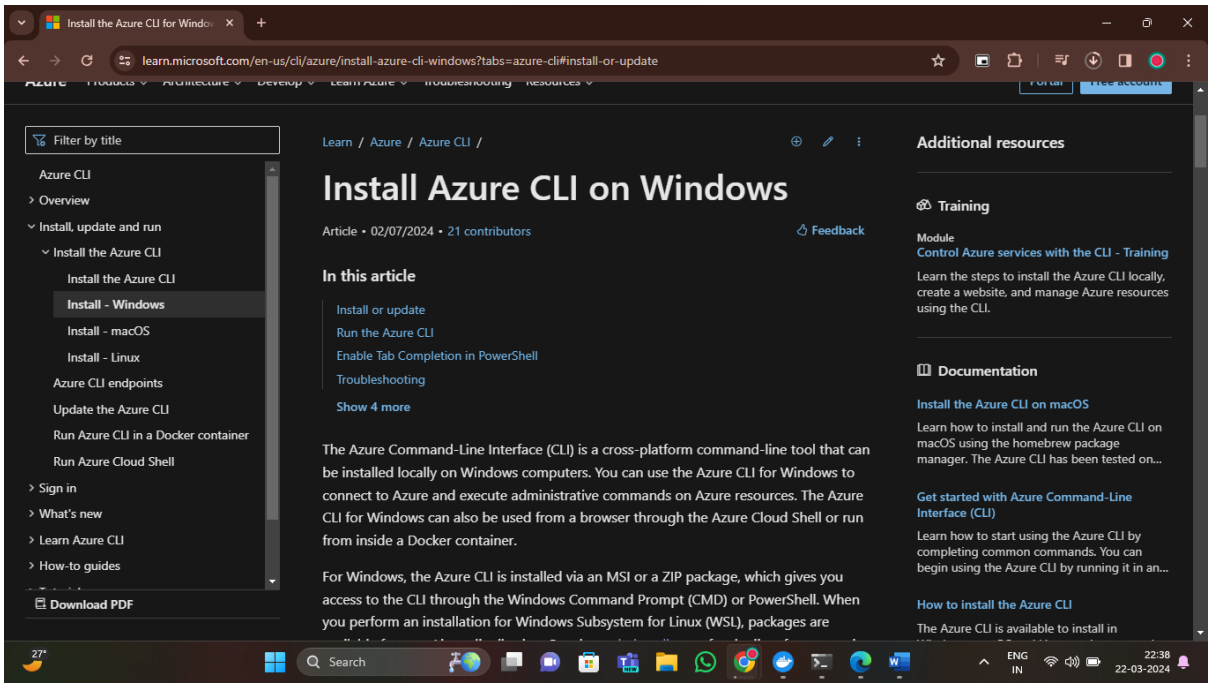
Step – 1: Install Node.js from https://nodejs.org/en



Step – 2: Install visual studio code from https://code.visualstudio.com/download



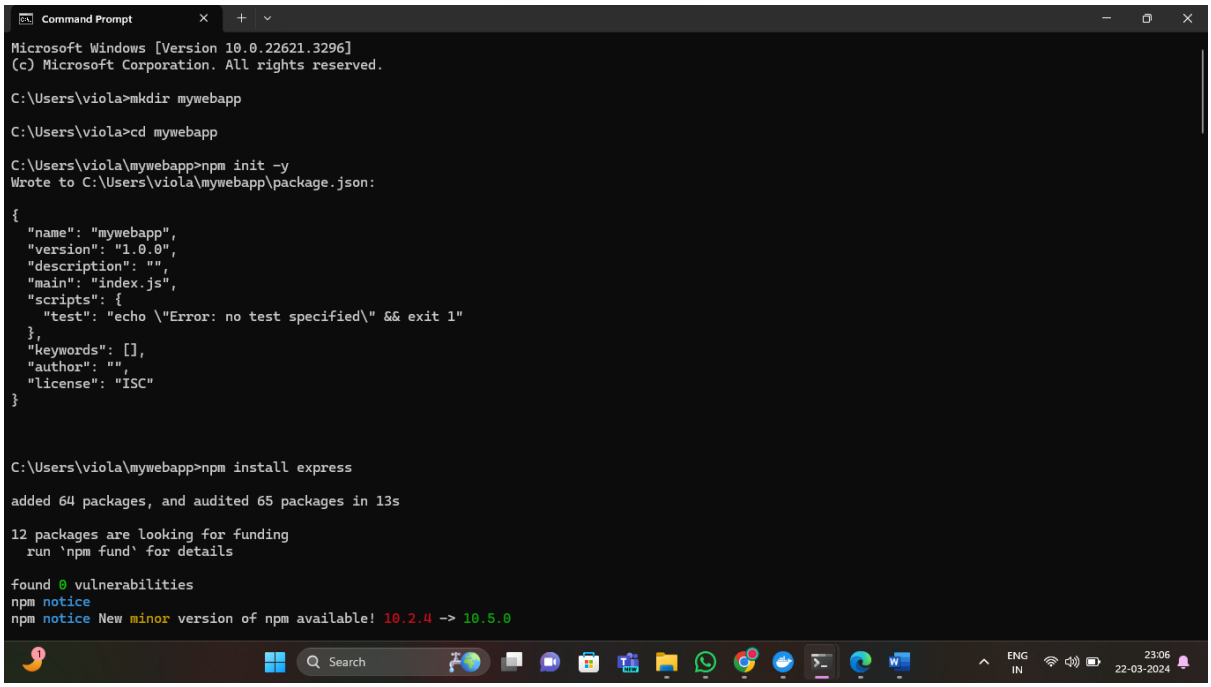
Step – 3: Install Azure from https://learn.microsoft.com/en-us/cli/azure/install-azure-cli-windows?tabs=azure-cli#install-or-update

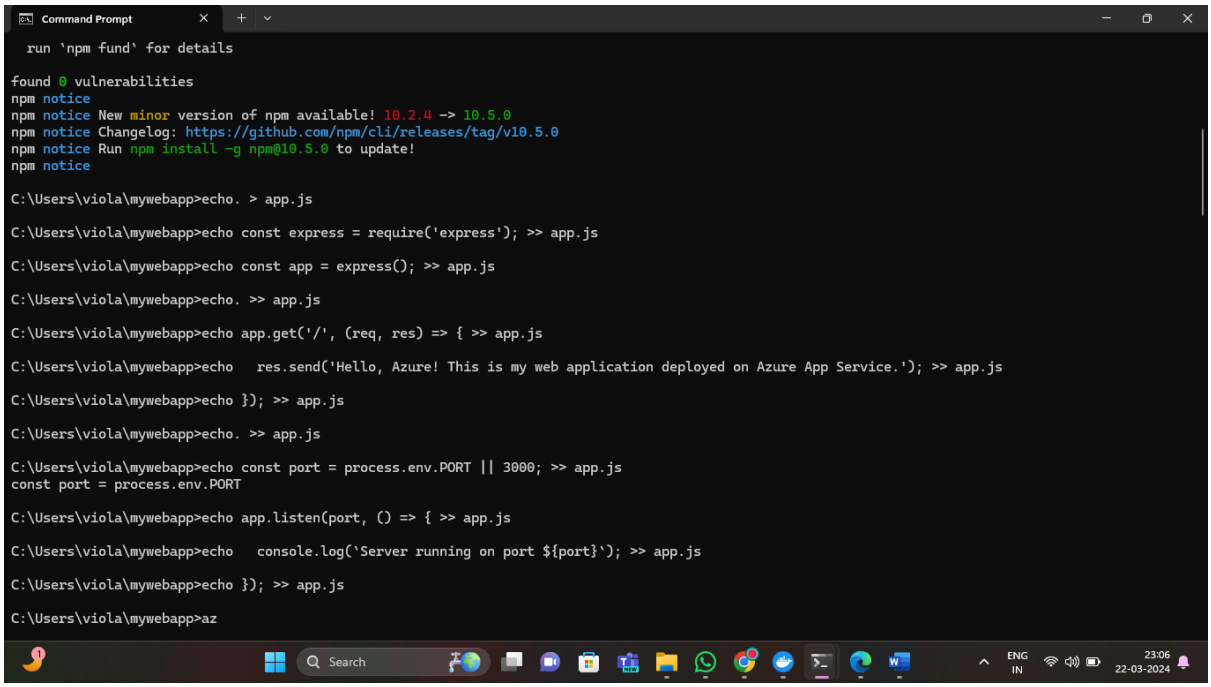


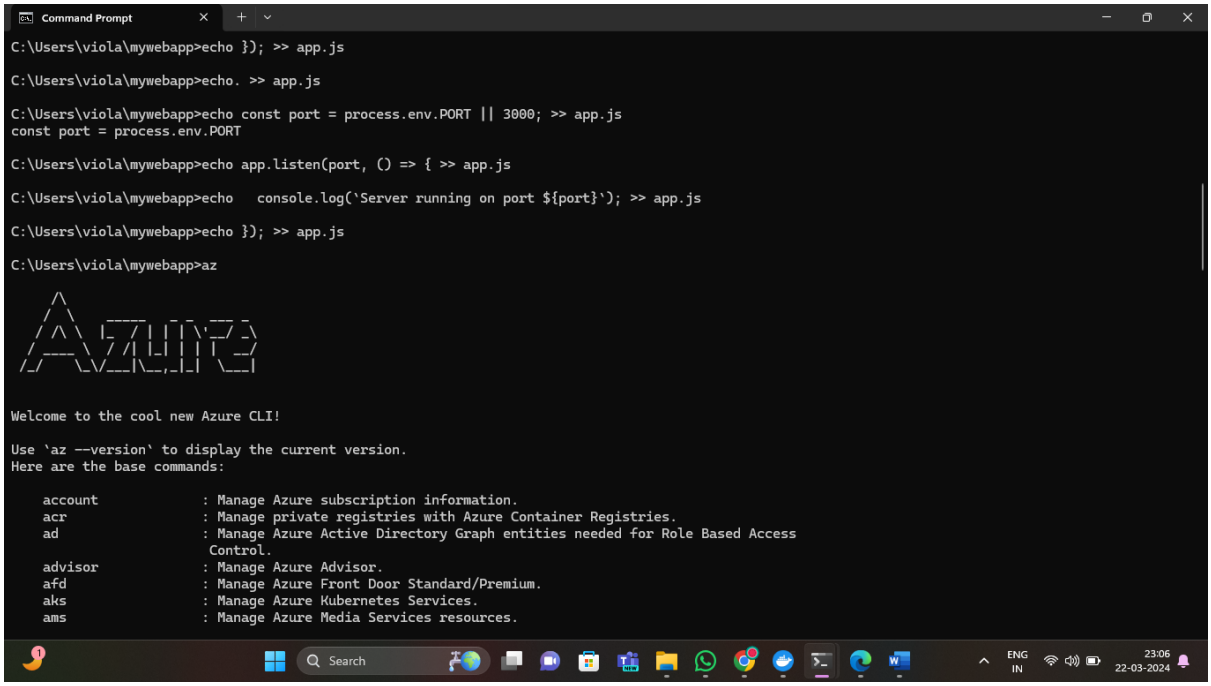
Code:

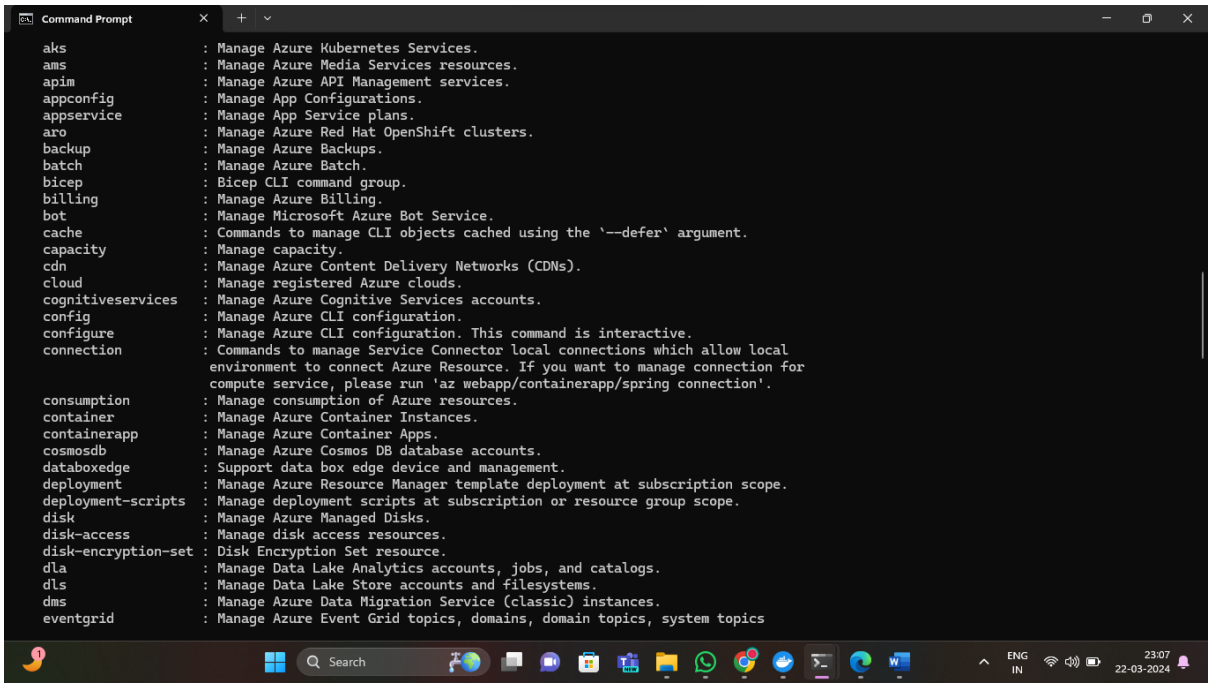
```
mkdir mywebapp

cd mywebapp

npm init -y

npm install express

const express = require('express');

const app = express();


app.get('/', (req, res) => {

  res.send('Hello, Azure! This is my web application deployed on Azure App Service.');

});


const port = process.env.PORT || 3000;

app.listen(port, () => {

  console.log(`Server running on port ${port}`);
```

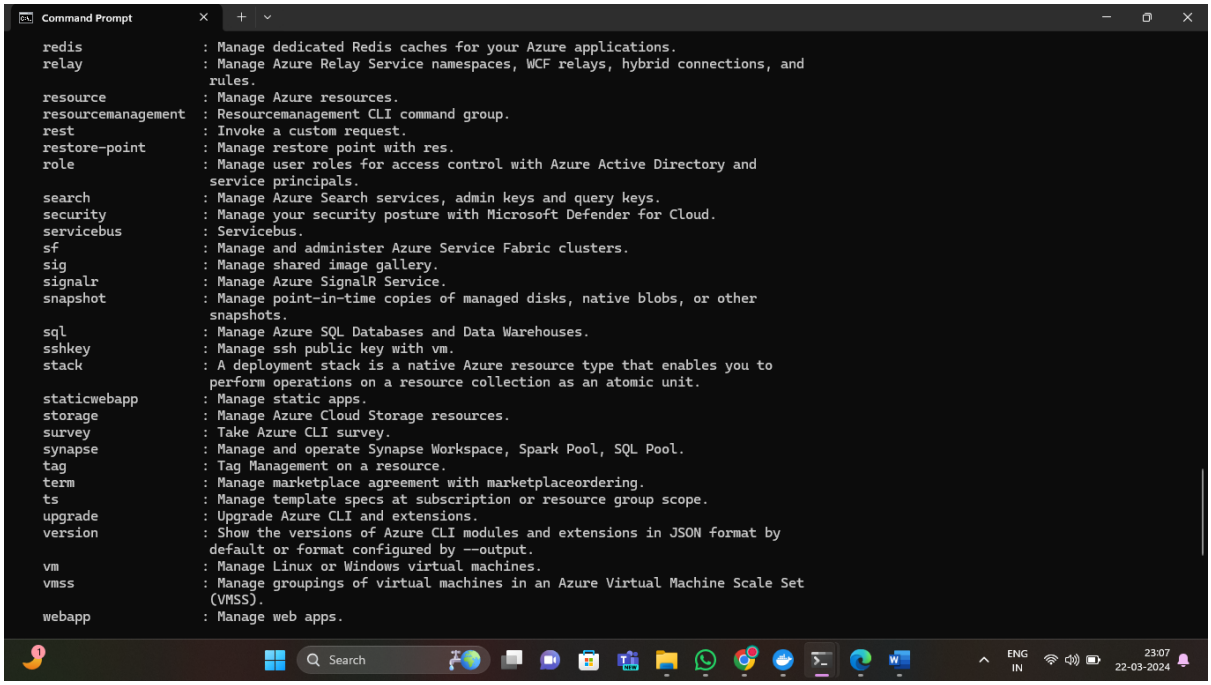
});


Command Prompt
Microsoft Windows [Version 10.0.22621.3296]
(c) Microsoft Corporation. All rights reserved.
C:\Users\viola>mkdir mywebapp
C:\Users\viola>cd mywebapp
C:\Users\viola\mywebapp>npm init -y
Wrote to C:\Users\viola\mywebapp\package.json:
{
"name": "mywebapp",
"version": "1.0.0",
"description": "",
"main": "index.js",
"scripts": {
"test": "echo \"Error: no test specified\" && exit 1"
},
"keywords": [],
"author": "",
"license": "ISC"
}
C:\Users\viola\mywebapp>npm install express
added 64 packages, and audited 65 packages in 13s
12 packages are looking for funding
run `npm fund` for details
found 0 vulnerabilities
npm notice
npm notice New minor version of npm available! 10.2.4 -> 10.5.0
Search
ENG
IN
23:06
22-03-2024


Command Prompt
run `npm fund` for details
found 0 vulnerabilities
npm notice
npm notice New minor version of npm available! 10.2.4 -> 10.5.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.5.0
npm notice Run npm install -g npm@10.5.0 to update!
npm notice
C:\Users\viola\mywebapp>echo. > app.js
C:\Users\viola\mywebapp>echo const express = require('express'); >> app.js
C:\Users\viola\mywebapp>echo const app = express(); >> app.js
C:\Users\viola\mywebapp>echo. >> app.js
C:\Users\viola\mywebapp>echo app.get('/', (req, res) => { >> app.js
C:\Users\viola\mywebapp>echo res.send('Hello, Azure! This is my web application deployed on Azure App Service.'); >> app.js
C:\Users\viola\mywebapp>echo }); >> app.js
C:\Users\viola\mywebapp>echo. >> app.js
C:\Users\viola\mywebapp>echo const port = process.env.PORT || 3000; >> app.js
const port = process.env.PORT
C:\Users\viola\mywebapp>echo app.listen(port, () => { >> app.js
C:\Users\viola\mywebapp>echo console.log(`Server running on port ${port}`); >> app.js
C:\Users\viola\mywebapp>echo }); >> app.js
C:\Users\viola\mywebapp>az
Search
ENG
IN
23:06
22-03-2024


Command Prompt
C:\Users\viola\mywebapp>echo }); >> app.js
C:\Users\viola\mywebapp>echo. >> app.js
C:\Users\viola\mywebapp>echo const port = process.env.PORT || 3000; >> app.js
const port = process.env.PORT
C:\Users\viola\mywebapp>echo app.listen(port, () => { >> app.js
C:\Users\viola\mywebapp>echo console.log(`Server running on port ${port}`); >> app.js
C:\Users\viola\mywebapp>echo }); >> app.js
C:\Users\viola\mywebapp>az
Welcome to the cool new Azure CLI!
Use `az --version` to display the current version.
Here are the base commands:
account : Manage Azure subscription information.
acr : Manage private registries with Azure Container Registries.
ad : Manage Azure Active Directory Graph entities needed for Role Based Access Control.
advisor : Manage Azure Advisor.
afd : Manage Azure Front Door Standard/Premium.
aks : Manage Azure Kubernetes Services.
ams : Manage Azure Media Services resources.
Search
ENG IN
23:06 22-03-2024


Command Prompt
aks : Manage Azure Kubernetes Services.
ams : Manage Azure Media Services resources.
apim : Manage Azure API Management services.
appconfig : Manage App Configurations.
appservice : Manage App Service plans.
aro : Manage Azure Red Hat OpenShift clusters.
backup : Manage Azure Backups.
batch : Manage Azure Batch.
bicep : Bicep CLI command group.
billing : Manage Azure Billing.
bot : Manage Microsoft Azure Bot Service.
cache : Commands to manage CLI objects cached using the `--defer` argument.
capacity : Manage capacity.
cdn : Manage Azure Content Delivery Networks (CDNs).
cloud : Manage registered Azure clouds.
cognitiveservices : Manage Azure Cognitive Services accounts.
config : Manage Azure CLI configuration.
configure : Manage Azure CLI configuration. This command is interactive.
connection : Commands to manage Service Connector local connections which allow local environment to connect Azure Resource. If you want to manage connection for compute service, please run 'az webapp/containerapp/spring connection'.
consumption : Manage consumption of Azure resources.
container : Manage Azure Container Instances.
containerapp : Manage Azure Container Apps.
cosmosdb : Manage Azure Cosmos DB database accounts.
databoxedge : Support data box edge device and management.
deployment : Manage Azure Resource Manager template deployment at subscription scope.
deployment-scripts : Manage deployment scripts at subscription or resource group scope.
disk : Manage Azure Managed Disks.
disk-access : Manage disk access resources.
disk-encryption-set : Disk Encryption Set resource.
dla : Manage Data Lake Analytics accounts, jobs, and catalogs.
dls : Manage Data Lake Store accounts and filesystems.
dms : Manage Azure Data Migration Service (classic) instances.
eventgrid : Manage Azure Event Grid topics, domains, domain topics, system topics
Search
ENG IN
23:07 22-03-2024



##Azure is Paid subscription so can’t proceed anymore.