**Part A: Mongo Atlas – Create a New Project and Database**

1. Create a new project under the organization in your Mongo Atlas Account. Click **Next** after enter the project name, and click the **Create Project** button.

A screenshot of a computer

Description automatically generated

Figure A-1: Project Creation

1. Once the program creation completed, setup the deployment from the dashboard by clicking the **Create** button as in Figure 2.

A screenshot of a cloud computing application

Description automatically generated

Figure A-2: Create a Deployment

1. Select the M0 server for the free tier and click the **Create** button to proceed. You may choose the region closer to Malaysia.

A screenshot of a computer

Description automatically generated

Figure A-3: Deployment

1. **Create** the default username and password to connect to the database.

A screenshot of a computer

Description automatically generated

Figure A-4: User Authentication

1. Add an entry with the IP address of **0.0.0.0** to allow access to the database from anywhere for the ease of development.

A screenshot of a computer

Description automatically generated

Figure A-5: Network Access List

1. Once the process completed, you should be able to see the database created successfully in the dashboard.

A screenshot of a software update

Description automatically generated

Figure A-6: Completion

**Part B: Github Student Account**

1. Goto <https://education.github.com/benefits?type=student> to create a new Github account using your UTeM Email Address.

**A screenshot of a screen

Description automatically generated**

Figure B-1: Github Student Account

1. After you successfully verify your education account, activate the Copilot from <https://github.com/features/copilot>.

A screenshot of a computer program

Description automatically generated

Figure B-2: Copilot Activation

1. Create a new repository, select a unique repository name for your project.
2. Make the repository **Public**, Add a **README** file, and finally Add .gitignore to **Node**.
3. Click the **Create Repository** button to continue.

A screenshot of a computer

Description automatically generated

Figure B-3: Create Repository

**Part C: Microsoft Azure – Create a Dreamspark Account**

1. Browse to <https://azure.microsoft.com/free/students> to create a student account in Azure using your UTeM Email Address.
2. Fill in the registration form and click the Verify Academic Status complete.

A screenshot of a computer

Description automatically generated

Figure C-1

1. Go to <https://portal.azure.com/#allservices>, click **+** to create a new **Web App**.

A screenshot of a computer

Description automatically generated

Figure C-2: Azure Service List

1. Choose a unique name for the **Web App**. Change the Runtime Stack to **Node 18 LTS**, and the Operating System to **Linux.** Then, select the region closer to Malaysia.
2. Click **Review + Create** button to continue.

A screenshot of a black screen

Description automatically generated

Figure C-3: Basic Info

1. Review the summary generated, click the **Create** button the proceed.
2. Wait for the deployment completed, the web app should be listed in the dashboard.

A screenshot of a computer

Description automatically generated

Figure C-4: Summary

**Part D: Microsoft Azure – CI/CD Development**

1. Click the Web App on the dashboard to check on its status and detail.

A screenshot of a computer

Description automatically generated

Figure D-1: Status of the Web App created

1. Go to the **Deployment Center** to setup the CI/CD with Github.

A screenshot of a computer

Description automatically generated

Figure D-2: Deployment Source

1. If you signed in to **Github** in *Step B* previously, the account will be automatically binded. Otherwise, proceed the sign in into **Github**.
2. From the Organization, select the one your create in Github, which should be the same as your account.
3. Select the repository your created in *Step B-5*.
4. Lastly, choose the Branch that you want to link to the repository.
5. Click the Save button on the toolbar to continue.

A screenshot of a computer

Description automatically generated

Figure D-3: Github Deployment Setup

1. After the process completed, you should able to a new commit was added to your repository in the Github.

A screenshot of a computer

Description automatically generated

Figure D-4: Repository Commits

**Part E: Express Framework**

1. Clone the repository into your computer. Then open the folder using VSCode.

A screenshot of a computer

Description automatically generated

1. Open the terminal and enter the npm init, accept all the default value and complete the process.
2. Install the Express Framework with npm install express --save.
3. Create a file index.js in the project folder, and paste the following code:

const express = require('express')

const app = express()

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

app.use(express.json())

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

res.send('Hello World!')

})

app.listen(port, () => {

console.log(`Example app listening on port ${port}`)

})

1. Open the package.json file, edit the scripts to as follow:

"scripts": {

"start": "node index.js"

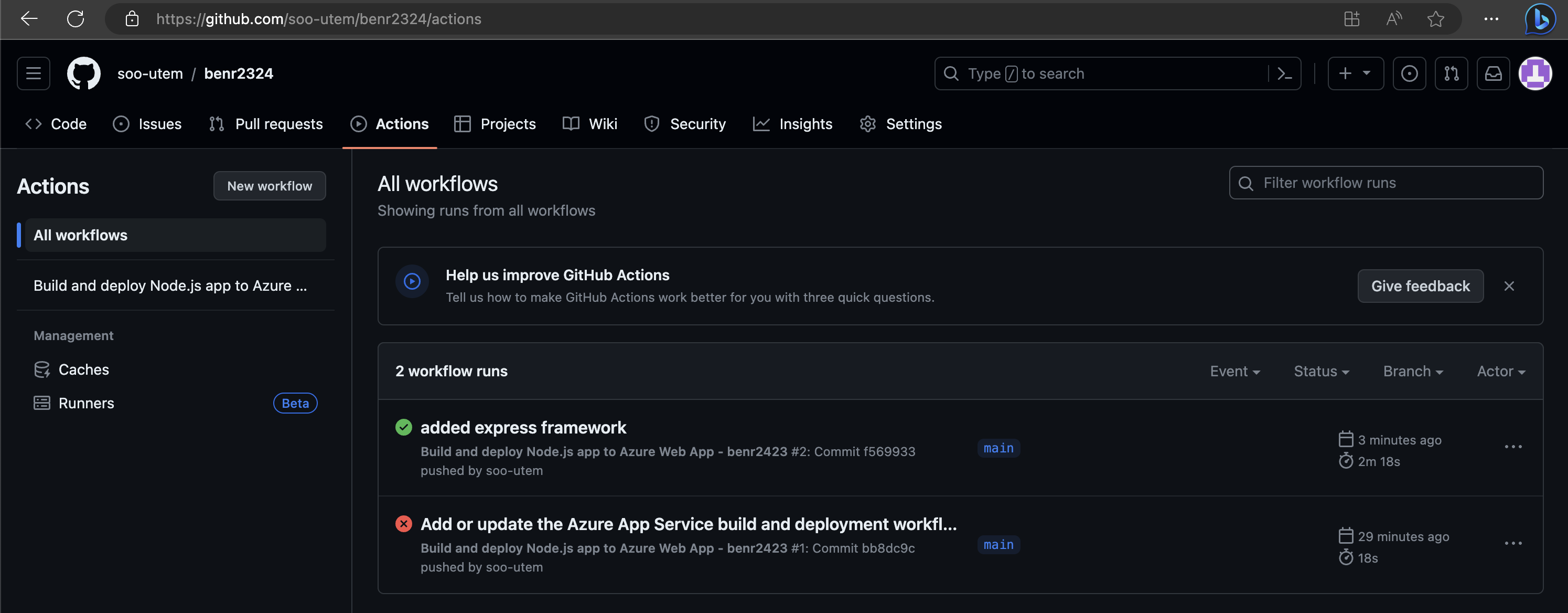
},

1. Save all the files and push a commit to the repository.

A screen shot of a computer

Description automatically generated

1. Go to the Github repository, check on the **Actions** tab to view the CI/CD process for the newly commit change.



1. Open the Web App address from a browser to verify the result.

A screenshot of a computer

Description automatically generated