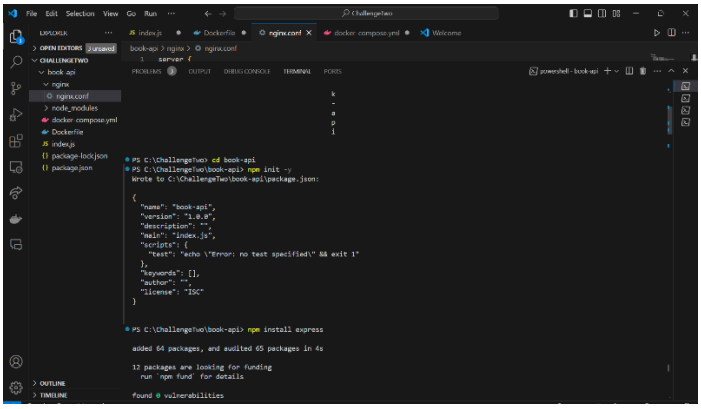
Challenge 2 :

Step by Step Instructions :

1. For this challenge firstly I created folder namely book-api in which I initialized a new node.js project by using “npm init -y”.
2. After that I installed express using “npm install express”. I was learning express so thought to try it in this project !



3. Then in my book-api folder I created new file named “index.js” in which I added name of some of my favorite books .

```
const express = require('express');

const app = express();
const books = [
    { id: 1, title: "Who will cry when you die", author: "Robin Sharma" },
    { id: 2, title: "The Monk who sold his ferrari", author: "Robin Sharma" },
    { id: 3, title: "The Secret", author: "Rhonda Byrne" },
    { id: 4, title: "Be Water my Friend" , author: "Bruce Lee"}
];
app.get('/api/books', (req, res) => {
    res.json(books);
});
app.get('/api/books/:id', (req, res) => {
    const book = books.find(b => b.id === parseInt(req.params.id));
    if (!book) return res.status(404).send('Book is not found !!!!');
    res.json(book);
});
const PORT = process.env.PORT || 3000;
app.listen(PORT, () => {
    console.log(`Server is running on the port -> ${PORT}`);
});
```

4. Then in my book-api folder I created the Dockerfile. In which I used WORKDIR, COPY and RUN as new commands than my Challenge one.
   WORKDIR is used to alter working directory.
   COPY instruction copies files and directories.

RUN command is used to execute build commands.
Used https://docs.docker.com/reference/dockerfile/ website for research.

```
FROM node:14
WORKDIR /usr/src/app
COPY package*.json ./
RUN npm install
COPY . .
CMD [ "node", "index.js" ]
EXPOSE 3000
```

5. On some research I got to know that we need to configure Nginx as a reverse proxy so that it could forward requests to the Node.js application.
6. After it I created a new folder named nginx and inside it I created nginx.conf in it and added Reverse Proxy.
   Used https://docs.nginx.com/nginx/admin-guide/web-server/reverse-proxy/ for help.

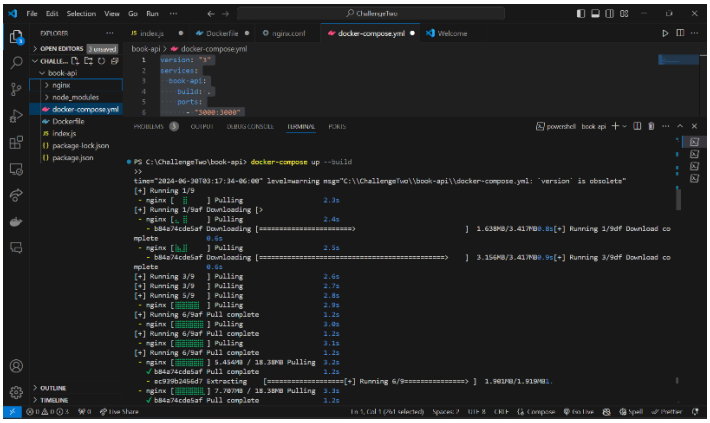
```
server {
    listen 8080;
    location /api {
        proxy_pass http://book-api:3000;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}
```

7. And lastly so that I could define and manage multi-container Docker applications and linkage of Node.js app with Nginx I created a docker compose file.
8. For it firstly I created docker-compose.yml file in our book-api and then added the below content with the help of https://docs.docker.com/compose/ .

```
version: "3"
services:
  book-api:
    build: .
    ports:
      - "3000:3000"
  nginx:
    image: nginx:alpine
    volumes:
      - ./nginx/nginx.conf:/etc/nginx/conf.d/default.conf
    ports:
      - "8080:8080"
    depends_on:
```

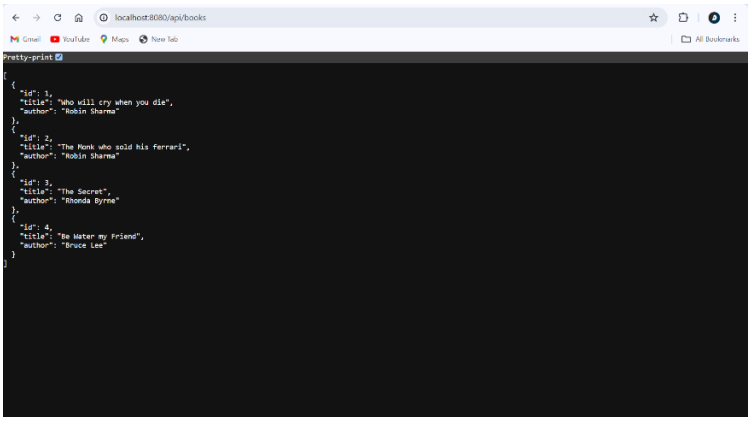
- book-api

9. And to build Docker images and start the containers as defined in the docker-compose.yml I opened Terminal -> New Terminal and then typed “docker-compose up –build”



10. At last checking “http://localhost:8080/api/books” and “http://localhost:8080/api/books/1” on our local browser for confirmation.

11. http://localhost:8080/api/books



12. http://localhost:8080/api/books/1