### Name: IKIREZI Joy ID:24885

### Docker and Serverless Deployment Assessment Report

**Book-app Application**

### 1. Introduction to the Application

**Book App** is a simple platform for tracking and managing your book collection. It allows users to add, view, and delete books with ease. Whether you're an avid reader or just getting started, this app helps you stay organized and keep track of your literary journey.

Key Features:

* Add books to your collection with title, author, and genre details.
* View a list of all books you've added.
* Delete books from your collection when you're done reading.
* Update book information to keep your collection accurate.

#### Technical Stack:

* **Frontend:** HTML, CSS
* **Backend:** Node.js with Express.js
* **Database:** MongoDB Atlas (cloud-based database)
* **Containerization:** Docker
* **Deployment:** Render.com

### 2. Steps to Dockerize the Application

#### 2.1 Creating the Dockerfile

A **Dockerfile** was created in the root directory with the following configuration:

#### FROM node:14

#### WORKDIR /app

#### COPY package\*.json ./

#### RUN npm install

#### COPY . .

#### EXPOSE 3000

#### CMD ["npm", "start"]

#### 2.2 Creating .dockerignore

A **.dockerignore** file was added to exclude unnecessary files:

#### node\_modules/

#### .env

#### \*.log

#### test/

#### .vscode/

#### Dockerfile

#### .git/

#### 2.3 Testing Docker Container Locally

The following commands were executed to test the containerized application:

docker build -t book-app .

docker run -p 3000:3000 book-app

### 3. Deployment Steps on Render.com

#### 3.1 Repository Setup:

* Created a public GitHub repository: <https://github.com/Joy4-hub/DevOps-Assignment.git>
* Pushed application code, including Dockerfile

#### 3.2 Render.com Configuration:

* Created a new account on Render.com
* Selected **Web Service** deployment
* Connected GitHub repository
* Configured deployment settings:
  + **Runtime Environment:** Docker
  + **Dockerfile Path:** Dockerfile
  + **Branch:** main
  + **Instance Type:** Free tier

#### 3.3 Environment Configuration:

* Added required environment variables:
* **MONGODB\_URI:** mongodb+srv://joykhagarama04:xXeOZGbvGECmNHLu@cluster0.rsgdq.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0
* **PORT:** 3000

#### 3.4 Deployment Verification:

* Monitored build logs for successful deployment
* Tested application functionality on the deployed URL
* Verified database connectivity

### 4. Challenges and Solutions

#### 4.1 Docker Configuration Challenge:

* **Issue:** Initial Docker build failed due to missing Dockerfile
* **Solution:** Created and properly configured Dockerfile with correct path specification in Render
* **Learning:** Importance of proper file naming and path specification in Docker deployments

#### 4.2 Environment Variables Management:

* **Issue:** Securing sensitive database credentials
* **Solution:** Utilized Render's environment variables feature for secure credential management
* **Learning:** Best practices for handling sensitive configuration in containerized applications

#### 4.3 Database Connectivity:

* **Issue:** Ensuring proper MongoDB Atlas connection from containerized environment
* **Solution:** Configured MongoDB Atlas network access to allow connections from any IP
* **Learning:** Understanding cloud database security and access management

### 5. Access Information

#### 5.1 Deployed Application URL:

#### <https://devops-assignment-9brj.onrender.com>

#### 5.2 GitHub Repository:

### <https://github.com/Joy4-hub/DevOps-Assignment.git>

### 6. Conclusion

The successful deployment of the Book App demonstrates the effective use of Docker for containerization and Render for serverless deployment. The application continues to function seamlessly while running in a cloud environment. This highlights the practical application of Docker and cloud deployment concepts for building scalable and efficient web applications.