**Dokumentation ToDo-Applikation**

**Klonen des Repositories**

Ziel: Eine lokale Kopie des Projekts erstellen, um daran arbeiten zu können.

**Schritt-für-Schritt-Anleitung:**

1. Besuche das GitHub-Repository „docker-nodejs-sample“ und klicke auf Fork, um eine Kopie in deinem eigenen GitHub-Konto zu erstellen.
2. Öffne Git Bash und führe den folgenden Befehl aus:

“git clone https://github.com/MeinBenutzername/docker-nodejs-sample.git”

“cd docker-nodejs-sample”

Eine zweite lösung wäre über github desktop ganz einfach zu klonen.

Erklärung:

Dieser Schritt erstellt eine vollständige Kopie des Repository-Codes, damit du ihn lokal ändern kannst.

**Erstellung der README.md**

Ziel: Eine leicht verständliche Anleitung für die Installation und Nutzung des Projekts im Markdown-Format zu erstellen.

Schritt-für-Schritt-Anleitung:

1. Erstelle im Root-Verzeichnis des Projekts eine Datei namens README.md.

2. Füge Informationen über die Installationsschritte hinzu, wie:

Ein Bild, das Text, Screenshot, Software enthält.

Automatisch generierte BeschreibungKlonen des Repositories

Installation der Pakete

Docker-Konfiguration und -Installation

Start der Anwendung im Docker-Container

Erklärung:

Die README-Datei soll dir und anderen Nutzern ermöglichen, das Projekt schnell und einfach zu installieren und zu starten.

**Verwendung von Git (Commit, Push)**

Ziel: Änderungen regelmäßig sichern und auf GitHub synchronisieren.

Schritt-für-Schritt-Anleitung:

1.Füge neue oder geänderte Dateien zur Git-Überwachung hinzu:

“git add . ”

2.Erstelle eine Commit-Nachricht, um die Änderungen zu dokumentieren:

“git commit -m” "Beschreibe die Änderung, z.B. Add README.md with installation instructions"

3.Push die Änderungen in dein GitHub-Repository:

“git push origin main”

Erklärung:

Durch regelmäßiges Committen und Pushen sicherst du deinen Fortschritt und hältst die Historie der Änderungen fest.

**Erstellung und Nutzung von Docker-Containern**

Ziel: Die Anwendung in einem Docker-Container ausführen, um sie in einer isolierten Umgebung betreiben zu können.

Ein Bild, das Elektronik, Text, Screenshot, Display enthält.

Automatisch generierte Beschreibung**Schritt-für-Schritt-Anleitung:**

1. Erstelle eine leere Dockerfile mit dem Befehl:

“touch Dockerfile”

2. Baue das Docker-Image und tagge es als todo-app:

“docker build -t todo-app . ”

3. Nachdem der Build abgeschlossen ist, starte die Anwendung mit:

“docker run -p 3000:3000 todo-app”

4. Die Anwendung läuft auf Port 3000.

* Erklärung: Diese Schritte zeigen den Aufbau und Start des Containers für die Anwendung. Die Ausgabe zeigt, dass die Anwendung auf Port 3000 läuft und die SQLite-Datenbank verwendet wird, um Daten zu speichern.

Das Endergebnis sieht dann so aus:

Ein Bild, das Text, Screenshot, Software, Computersymbol enthält.

Automatisch generierte Beschreibung