

Aufgabe: Eine einfache Grußanwendung und GitHub Actions Workflow

Die Grußanwendung (Greeting App) ist eine einfache Webanwendung, die es Benutzern ermöglicht, ihren Namen einzugeben und dann einen personalisierten Gruß zu erhalten.

1. HTML-Struktur:

- Die Anwendung besteht aus einer HTML-Datei (`greeting.html`), die die grundlegende Struktur der Webseite definiert. Es gibt ein Eingabefeld für den Namen, einen Button zum Absenden des Namens, eine Überschrift, einen Abschnitt für die Grußnachricht und einen allgemeinen Begrüßungstext.

2. Eingabefeld und Button:

- Ein Texteingabefeld mit der ID `nameInput` ermöglicht es Benutzern, ihren Namen einzugeben.
- Ein Button mit einem `onclick` Attribut, das die JavaScript-Funktion `submitName()` aufruft, wird verwendet, um den eingegebenen Namen zu verarbeiten.

3. JavaScript-Funktionen (`greetingApp.js`):

- In der JavaScript-Datei sind zwei Funktionen definiert: `submitName` und `updateUI` .
- Die Funktion `submitName` wird aufgerufen, wenn der Benutzer auf den "Submit"-Button klickt. Sie liest den eingegebenen Namen, erstellt eine personalisierte Grußnachricht und ruft dann die Funktion `updateUI` auf.
- Die Funktion `updateUI` aktualisiert den Textinhalt des Elements mit der ID `greetingMessage` mit der erstellten Grußnachricht.

4. Grußnachricht:

- Die Grußnachricht wird dynamisch generiert und enthält den eingegebenen Namen des Benutzers. Zum Beispiel könnte die Grußnachricht lauten: "Hello, [Name]!"

5. Allgemeine Begrüßungsnachricht:

- Ein allgemeiner Begrüßungstext unterhalb des Grußnachricht-Abschnitts lautet: "Welcome to our Greeting App."

6. Einbindung von JavaScript:

- Das JavaScript-Datei (`greetingApp.js`) wird über ein `<script>` Element in die HTML-Seite eingebunden.

Die Gesamtfunktionalität der Grußanwendung besteht darin, dass Benutzer ihren Namen in das Eingabefeld eingeben, auf den "Submit"-Button klicken und dann eine personalisierte Grußnachricht erhalten. Die Anwendung gibt eine freundliche Begrüßung aus und zeigt sie auf der Webseite an.

Schritt 1: Lokales Projekt erstellen

▼ Projektverzeichnis erstellen und wechseln:

```
mkdir greeting-app
cd greeting-app
```

▼ HTML-Datei erstellen (`greeting.html`):

```
<!DOCTYPE html><html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Greeting App</title>
</head>
<body>
  <h1>Greeting App</h1>
  <label for="nameInput">Enter your name:</label>
  <input type="text" id="nameInput" name="nameInput">
  <button onclick="submitName()">Submit</button>
  <h2 id="greetingMessage"></h2>
  <p>Welcome to our Greeting App.</p>
```

```
<script src="greetingApp.js"></script>
</body>
</html>
```

▼ JavaScript-Datei erstellen (`greetingApp.js`):

```
function submitName() {
  const enteredName = document.getElementById('nameInput').value;
  const greetingMessage = `Hello, ${enteredName}!`;
  updateUI(greetingMessage);
}

function updateUI(message) {
  document.getElementById('greetingMessage').innerText = message;
}

module.exports = { submitName, updateUI };
```

1. Lokale Tests durchführen:

Öffne die `greeting.html` im Browser und überprüfe, ob die Grußanwendung wie erwartet funktioniert.

Schritt 2: GitHub-Repository erstellen

▼ GitHub-Repository erstellen:

- Gehe zu [GitHub](#) und logge dich ein.
- Klicke auf den "+"-Button in der rechten oberen Ecke und wähle "New repository" aus.
- Gib einen Repository-Namen ein, z.B., "greeting-app".
- Füge eine optionale Beschreibung hinzu und lasse die Optionen "Public" und "Initialize this repository with a README" ausgewählt.
- Klicke auf "Create repository".

Schritt 3: Lokales Git-Repository einrichten

▼ Lokales Git-Repository einrichten:

- Öffne dein Terminal oder die Kommandozeile.
- Navigiere zum Verzeichnis deiner Grußanwendung.
- Führe die folgenden Befehle aus, um dein lokales Git-Repository einzurichten und die Dateien hinzuzufügen:

```
git init
git remote add origin <URL-deines-neuen-Repositories>
git add .
git commit -m "Initial commit"
```

Ersetze `<URL-deines-neuen-Repositories>` durch die tatsächliche URL deines neuen GitHub-Repositories.

Schritt 4: GitHub Actions-Workflow erstellen

1. GitHub Actions-Workflow erstellen:

- Erstelle im Hauptverzeichnis deiner Grußanwendung eine Datei namens `.github/workflows/main.yml`, der den Code auscheckt.

2. Datei speichern.

Schritt 5: Dateien ins GitHub-Repository pushen

1. Dateien ins GitHub-Repository pushen:

```
git push -u origin main
```

Schritt 6: Überprüfung des Workflows auf GitHub

1. Workflow auf GitHub überprüfen:

- Gehe zu deinem GitHub-Repository.
- Klicke auf den "Actions"-Tab.

- Du solltest den gerade erstellten Workflow sehen. Klicke darauf, um die Details anzuzeigen.
- Überprüfe, ob der Workflow automatisch gestartet wurde und ob alle Schritte erfolgreich abgeschlossen wurden.