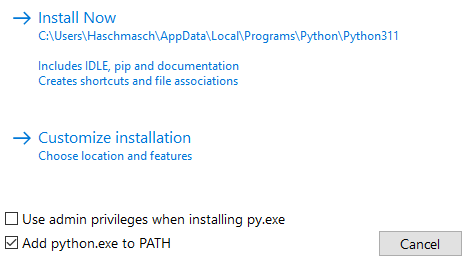
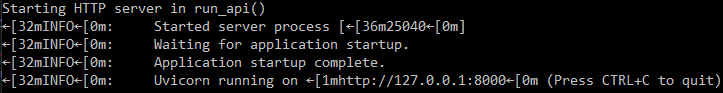
# Anleitung API

1. Python 3.11 installieren (<https://www.python.org/downloads/release/python-3116/>)
   1. Zusätzlich Python als Umgebungsvariable registrieren:  
      
   2. Wenn die Option nicht verfügbar ist, kann man den Python Installer nochmal ausführen und die Option nachträglich auswählen.
2. PostgreSql 15 oder 16 mit den Standarteinstellungen installieren (<https://www.postgresql.org/download/windows/>)
   1. Sicherstellen, dass der DB-Server im Hintergrund läuft.
3. Python-Module installieren:
   1. Mit der CMD mit „cd [PROJEKTPFAD]/WebEngineering2/API/“ in den Projektordner navigieren.
   2. „py -m pip install -r requirements.txt“ ausführen. Dann werden alle benötigten Module automatisiert installiert.
4. Im Dateiexplorer auf den Ordner „[PROJEKTPFAD]/WebEngineering2/API/Utils“ navigieren. Dort ist eine „config.json“ abgelegt.
5. Den ConnectionString auf den installierten Datenbank-Server anpassen. Der Name der Datenbank ist Standardmäßig auf „web2\_db“ festgelegt. Dieser kann geändert werden. Wenn die API gestartet wird, wird die Datenbank mit dem Namen im ConnectionString und den benötigten Tabellen automatisiert angelegt.
6. Die API starten, indem man sie über die CMD aufruft:
   1. Mit „cd [PROJEKTPFAD]/WebEngineering2/API“ in den API-Ordner des Projekts navigieren.
   2. Dann mit „py “[PROJEKTPFAD]/WebEngineering2/API/main.py““ die API starten.
   3. Die IP-Adresse und der Port der API werden dann in der CMD angezeigt:  
      
   4. Falls das Starten der API fehlschlagen sollte, kann eine mögliche Ursache sein, dass die API die Datenbank nicht anlegen konnte. Die Datenbank kann auch über ein SQL-Statement angelegt werden. Im Ordner „[PROJEKTPFAD]/WebEngineering2/DB\_Queries“ ist ein Skript mit dem Namen „web2\_db.sql“ zu finden. Den Inhalt dieses Skripts kann man kopieren und in einem Datenbank-Client ausführen, während man mit der Postgres-Datenbank verbunden ist. Danach sollte das starten der API funktionieren.
      1. Mit der Installation der Postgres-Datenbank sollte ein Datenbank-Client namens „pgAdmin“ zusätzlich installiert werden. Mit ihm kann das Skript ausgeführt werden.
7. Um die Funktionsfähigkeit der API zu testen kann die Dokumentation unter „[IP-ADRESSE]/docs“ im Browser aufgerufen werden.

**Hinweis:**  
Die Datenbank wird mit Standartwerten von der API initialisiert. Dabei werden die Kategorien, Subkategorien, 5 Angebote und ein User angelegt. Die Benutzerdaten des Users sind:

Name: Admin

Email: admin@example.com

Passwort: admin123

# Anleitung Frontend

1. Node.js auf der neusten Version installieren (<https://nodejs.org/en/download/current>)
2. Über cmd auf den Pfad „cd [PROJEKTPFAD]/WebEngineering2/Frontend“
3. „npm i“ oder „npm install“ in der Konsole eingeben
4. Zum Starten des Projekts „npm start“ in der Konsole eingeben (Achtung: die API muss vorher gestartet werden, da sonst Fehler auftreten)
5. Unter „<http://localhost:3000/>“ wird die Website dann gestartet (Es kann ein paar Minuten dauern, bis es geladen hat)

Bei Problemen oder Fragen kann man uns unter der folgenden Mail-Adresse erreichen:

rob.hassler.22@lehre.mosbach.dhbw.de