

# BVB Checker - PyInstaller Setup

## Dateien herunterladen und organisieren

Erstelle folgende Ordnerstruktur auf deinem Computer:

```
BVBChecker_Build/  
├── main.py  
├── rule_engine.py  
├── requirements.txt  
├── bvb_checker.spec  
├── build.py  
├── install.bat  
└── data/  
    └── diagnoseliste_corrected.csv
```

## Schritt 1: Python-Umgebung vorbereiten

```
bash  
  
# Erstelle virtuelle Umgebung (empfohlen)  
python -m venv bvb_env  
bvb_env\Scripts\activate # Windows  
# source bvb_env/bin/activate # macOS/Linux  
  
# Installiere Dependencies  
pip install -r requirements.txt
```

## Schritt 2: Executable erstellen

### Option A: Automatischer Build

```
bash  
  
python build.py
```

### Option B: Manueller Build

```
bash  
  
pyinstaller bvb_checker.spec --clean --noconfirm
```

## Schritt 3: Testen

```
bash
```

```
# Teste das Executable
```

```
dist\BVBChecker.exe
```

Das Programm sollte starten und automatisch den Browser öffnen auf <http://127.0.0.1:8000>.

## Schritt 4: Installer erstellen

Kopiere die Dateien für das Installations-Package:

```
BVBChecker_Release/
```

```
|— BVBChecker.exe    (aus dist/ Ordner)
```

```
|— install.bat
```

```
|— README.txt
```

## Schritt 5: Installation in Arztpraxen

### Für IT-Personal:

1. Package auf Ziel-PC kopieren
2. [install.bat](#) als Administrator ausführen
3. Desktop-Verknüpfung wird erstellt
4. Startmenü-Eintrag wird erstellt

### Für Ärzte:

1. Doppelklick auf "BVB Checker" Desktop-Icon
2. Browser öffnet sich automatisch
3. ICD-Codes eingeben und prüfen

## Wichtige Hinweise

### Systemanforderungen

- Windows 10/11 (64-bit)
- Mindestens 100 MB freier Speicherplatz
- Keine Internetverbindung erforderlich

### Sicherheit

- Executable ist nicht signiert
- Windows Defender könnte warnen
- IT-Teams sollten Hash whitelisten oder Code-Signierung hinzufügen

## Fehlerbehebung

### "Python nicht gefunden"

- Python 3.11+ installieren von python.org
- PATH-Variable prüfen

### PyInstaller Fehler

- Requirements.txt vollständig installieren
- Virtual Environment verwenden
- Als Administrator ausführen

### Executable startet nicht

- Antivirus temporär deaktivieren
- In anderem Ordner testen
- Windows Defender Ausnahme hinzufügen

## Performance-Optimierung

Für schnelleren Start, uncomment die COLLECT-Option in `bvb_checker.spec`:

- Erstellt Ordner statt einzelner .exe
- Schnellerer Start
- Einfacher zu debuggen

## Updates

Für neue Diagnoseliste:

1. `diagnoseliste_corrected.csv` ersetzen
2. `build.py` erneut ausführen
3. Neue .exe an Praxen verteilen

## Nächste Schritte für Software-Prod-Team

1. **Code-Signierung hinzufügen** für vertrauenswürdige Installation
2. **Auto-Update-Mechanismus** für neue KBV-Listen
3. **Logging-System** für Debugging in Produktionsumgebung
4. **MSI-Installer** für professionellere Distribution
5. **Backup-Mechanismus** für Konfiguration und Daten

## Support

Bei Problemen die `build.py` logs prüfen oder manual debugging:

```
bash
```

```
pyinstaller --debug=all bvb_checker.spec
```