

Namn: Amir Qurbani

Kurs: Examensarbete

Gitrepo: <https://github.com/amir-qurbani/lokal-ai-assistent.git>

1. Mål

Målet är att bygga en lokal AI-assistent i Python som kan läsa och analysera Markdown- och PDF-filer på datorn. Assistenten ska kunna hitta gamla eller inaktiva filer som inte används längre och hjälpa användaren att städa upp.

Syftet är att skapa en smart och enkel app som håller ordning på dokument och frigör plats på datorn.

2. Scope (omfattning)

Appen kommer att köras lokalt och hantera både Markdown-filer och PDF-filer.

För lagring används en SQLite-databas, vilket betyder att ingen molnlagring eller externa servrar behövs.

Programmet ska kunna identifiera gamla och inaktiva filer som inte längre används.

Ingår:

- Lokal körning av appen
- Filhantering för PDF och Markdown
- Lagring i SQLite-databas
- Identifiering av gamla eller inaktiva filer

Ingår inte:

- Molnlagring eller externa servrar
- Avancerad grafisk användargränssnitt
- Automatiska borttagningar utan bekräftelse

3. Milstolpar (veckovis plan)

Vecka	Mål / Leverans	Beskrivning
-------	----------------	-------------

Vecka 45	Projektplan + start på Python-projektet	Skriv klart projektplan v1, skapa GitHub-repo och sätta upp Python-miljön (venv, VS Code, SQLite). Börja bygga första delen av appen: enkel filhantering med <code>pathlib</code> som kan lista PDF-filer i en mapp.
Vecka 46	Filindexering till SQLite	Appen ska kunna söka igenom mappar, läsa filinformation (namn, storlek, datum) och spara metadata i databasen.
Vecka 47	Textutvinning (PDF & Markdown)	Programmet ska kunna läsa text från PDF och Markdown och spara textinnehållet i databasen.
Vecka 48	AI-sök (Embeddings)	Implementera enkel AI-sökning med lokala embeddings för att kunna ställa frågor och få träffar från dokument.
Vecka 49	Test, UI och demo	Göra tester, lägga till ett enkelt gränssnitt (CLI eller liten webbsida), skriva dokumentation och visa slutdemo.

4. Risker och hantering

Risk	Sannolikhet	Konsekvens	Åtgärd / Hantering
Python är nytt för mig	Hög	Kan ta längre tid att komma igång	Planerar extra tid för övning i början och testar varje del steg för steg
Tidsbrist eller sjukdom	Medel	Kan försena leveransen	Jobbar lite varje dag, inte allt på slutet, och tar backup ofta
Problem med filhantering eller PDF-läsning	Medel	Kan stoppa utvecklingen	Testar med små filer först och läser dokumentation / frågor handledare
SQLite-databas fel eller korrupt data	Låg	Tappar data	Lägger till loggning, använder try/except och sparar kopior
För mycket fokus på AI-del i slutet	Medel	Missar grundfunktioner	Gör färdigt grundfunktionerna först innan AI-del läggs till

5. Kvalitetssäkring (QA)

För att säkerställa kvalitet kommer jag att:

- Testa varje funktion separat (manuellt och med små automatiska tester i pytest).
- Logga fel med logging för att snabbt kunna felsöka.

- Kontrollera att databasen uppdateras korrekt efter varje indexering.
 - Be en klasskamrater, testa programmet och ge feedback.
 - Göra en slutlig testkörning innan demo där jag visar hela flödet: indexering, visning, borttagning.
-

6. Demo-plan (första sprintdemo)

Datum: Fredag 7 november

Syfte: Visa att projektet är planerat, att miljön fungerar och att jag har kommit igång med Python-inläringen.

Plan för demonstrationen:

- Förklara projektidén: en lokal AI-assistent för att hantera och rensa gamla filer.
- Visa att projektplanen är klar (mål, scope, risker och milstolpar).
- Berätta att jag har installerat Python och skapat en fungerande utvecklingsmiljö (VS Code och SQLite).
- Förklara att jag ännu inte startat själva appen eftersom jag fokuserat på att lära mig Python.