

Medsafe AI

AI-driven medical safety assistant

Project Overview:

MedSafe AI is an intelligent, AI-powered healthcare assistance platform designed to enhance medicine safety awareness, symptom understanding, and early risk identification through a unified web-based system. Built using Streamlit, Optical Character Recognition (OCR), fuzzy matching algorithms, curated medical databases, and large language models (LLMs), the platform assists users in safely navigating medication usage and basic health concerns in an educational, non-diagnostic manner.

MedSafe AI streamlines multiple healthcare safety workflows by enabling users to check potential drug–drug interactions, extract medicines and active salts from prescription images, receive symptom-based guidance, log post-medication experiences, and assess emergency risk levels. By combining OCR-based prescription analysis, intelligent medicine name matching, rule-based risk scoring, and AI-generated explanations, the system delivers structured, transparent, and user-friendly safety insights.

Through the integration of computer vision, natural language processing, fuzzy logic, and generative AI, MedSafe AI provides a reliable environment for medicine awareness, preventive health education, and early warning support. The platform is particularly valuable for patients, caregivers, and health-tech researchers seeking accessible tools for medication safety monitoring, symptom clarification, and risk-aware decision support without replacing professional medical consultation.

Scenario 1: Medication Safety, Interaction Awareness, and Prescription Understanding

Patients and caregivers often face challenges such as unclear prescriptions, unfamiliar medicine names, and a lack of awareness about potential drug–drug interactions. Misinterpretation of handwritten prescriptions or combining medicines without proper knowledge can lead to avoidable health risks. MedSafe AI addresses this challenge by providing an intelligent, end-to-end environment where users can input medicine names or upload prescription images and receive clear, structured safety insights in an educational and non-diagnostic manner.

For example, a patient prescribed multiple medicines by different doctors can use MedSafe AI's Medicine Interaction Checker to enter all medications at once. The system applies fuzzy matching to accurately identify medicines from its curated database and checks for known interaction warnings. Any detected interaction is presented clearly, along with a concise

AI-generated safety note that summarizes the risk in simple language. This eliminates guesswork and reduces reliance on incomplete internet searches or subjective assumptions.

In cases where prescriptions are difficult to read, users can upload an image of the prescription. MedSafe AI leverages OCR and generative AI to extract medicine names along with their active drugs or salts in a structured JSON format. This allows users to better understand what they are taking and cross-check safety information. By combining OCR, fuzzy logic, and AI-based summarization within an interactive Streamlit interface, MedSafe AI improves medication literacy, enhances patient confidence, and promotes safer medicine usage without replacing professional medical advice.

Scenario 2: Symptom Guidance, Side-Effect Monitoring, and Early Risk Awareness

Individuals experiencing new symptoms or unexpected side effects after taking medicines often struggle to determine whether their experience is normal, requires monitoring, or needs urgent attention. Accessing reliable, understandable health information without jumping to conclusions or self-diagnosis remains a significant challenge. MedSafe AI addresses this gap by offering a structured symptom guidance system, a side-effect monitoring module, and an emergency risk predictor—focused on education, awareness, and early warning support.

For instance, a user experiencing discomfort after taking a prescribed medicine can log their age, gender, medicines taken, dosage, and post-medication experience in the Side-Effect Monitor. MedSafe AI analyzes this information and generates a short, educational response highlighting possible contributing factors and one clear precaution to watch for. The tone remains informative and non-diagnostic, helping users make sense of their experience without inducing panic or false certainty.

Additionally, users describing ongoing symptoms can receive basic guidance through the Symptom & Doubt Solver, which combines rule-based advice with AI-enhanced explanations that include home remedies, lifestyle suggestions, breathing or yoga exercises, dietary tips, and warning signs. If symptoms suggest potential danger, the Emergency Risk Predictor assigns a transparent risk score based on predefined rules and highlights the urgency level with clear next-step guidance. By integrating symptom analysis, experience logging, and risk scoring into a single platform, MedSafe AI empowers users to monitor their health more responsibly, recognize red flags early, and seek timely medical help—supporting safer decision-making through accessible, AI-driven health education.

Architecture Overview:

MedSafe AI is a modular, AI-driven healthcare safety platform designed to provide medicine interaction awareness, prescription understanding, symptom guidance, and emergency risk assessment through a unified web-based system. Developed using **Streamlit** as the interactive

front-end, the platform integrates computer vision, fuzzy logic, rule-based risk analysis, and large language models to deliver safe, transparent, and educational health insights.

The system follows a layered architecture consisting of a **user-facing presentation layer**, a **core intelligence layer**, and a **supporting data & AI services layer**, each responsible for a distinct part of the workflow.

Presentation Layer (Streamlit Interface):

This layer provides a responsive, tab-based user interface that allows users to enter medicine names, upload prescription images, describe symptoms, log side effects, and view emergency risk scores. Streamlit manages user inputs, real-time feedback, alerts, metrics, and structured visual outputs, ensuring ease of use and accessibility.

Core Intelligence Layer:

This layer contains the primary functional modules of MedSafe AI:

- **Medicine Interaction Checker:** Uses fuzzy string matching (RapidFuzz) to identify medicines from user input and cross-reference them with a curated medicine database to detect known interaction warnings.
- **Prescription OCR Module:** Utilizes Tesseract OCR to extract raw text from prescription images, followed by LLM-based parsing to identify medicine names and active drug salts in structured JSON format.
- **Symptom & Doubt Solver:** Combines rule-based symptom advice with LLM-powered expansion to provide educational guidance, home remedies, lifestyle suggestions, and warning signs.
- **Side-Effect Monitor:** Captures post-medication experiences along with demographic context and generates concise, educational explanations highlighting possible contributing factors and precautions.
- **Emergency Risk Predictor:** Applies predefined medical safety rules to calculate a transparent risk score and classify urgency levels, supported by AI-generated explanations for clarity.

Data & AI Services Layer:

This layer supports the intelligence modules through:

- A **Medicine Database (MED_DB)** containing interaction information and medicine metadata
- **OCR Engine (Tesseract)** for text extraction from prescription images
- **Large Language Models (via Ollama – LLaMA 3)** for summarization, structured extraction, and natural-language explanations

All components are tightly integrated to ensure smooth data flow between user input, AI processing, and output visualization. By combining deterministic rules with generative AI in a controlled and ethical manner, MedSafe AI delivers a reliable, scalable, and extensible

architecture suitable for healthcare education, preventive safety tools, and AI-assisted decision support—while clearly maintaining boundaries from clinical diagnosis or treatment.

Architecture Overview

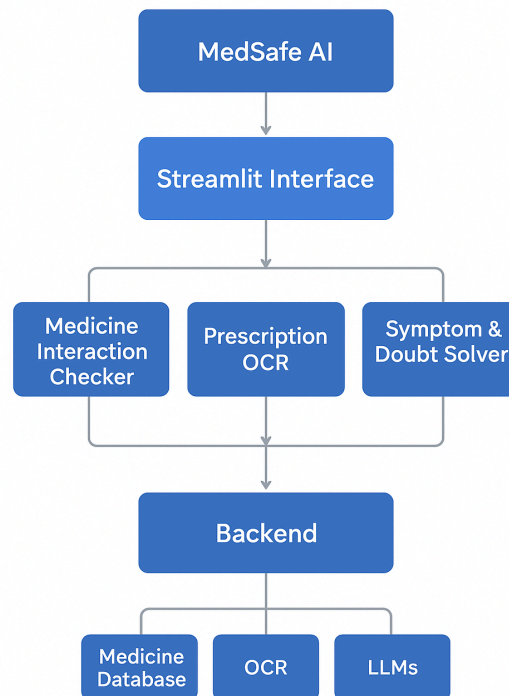


Figure 1: Architectural Overview of MedSafe AI

Core Technologies:

MedSafe AI is built using a carefully selected stack of modern web, AI, and data-processing technologies to deliver reliable medicine safety insights, prescription understanding, symptom guidance, and risk assessment. Each component plays a critical role in enabling accurate text extraction, intelligent matching, AI-assisted reasoning, and an interactive user experience while maintaining safety and transparency.

- **Streamlit (Frontend Framework)**

Provides a clean, real-time, and interactive web interface for all MedSafe AI modules. Streamlit manages tab-based navigation, user inputs, alerts, metrics, image previews, and expandable result sections, ensuring an accessible and responsive user experience without requiring complex frontend development.

- **Python (Core Programming Language)**

Acts as the backbone of the entire system, enabling seamless integration of OCR, fuzzy matching, rule-based logic, AI inference, and UI rendering. Python's extensive ecosystem supports rapid development, modular design, and maintainable code structure.

- **Tesseract OCR (Prescription Text Extraction)**

Used to extract raw textual content from uploaded prescription images. Tesseract enables MedSafe AI to process both printed and handwritten prescriptions, forming the foundation for medicine detection and further AI-based parsing.

- **RapidFuzz (Fuzzy Matching Engine)**

Handles approximate string matching to accurately identify medicine names even when OCR output is noisy or user input contains spelling variations. This improves robustness and reliability when mapping extracted text to known medicines in the database.

- **Medicine Database (MED_DB)**

A curated local data repository containing medicine names, metadata, and known interaction warnings. It serves as the authoritative reference for interaction checking and safety rule enforcement within the platform.

- **Large Language Models – LLaMA 3 (via Ollama)**

Power multiple AI-driven features, including interaction summarization, structured extraction of medicines and salts, symptom guidance expansion, side-effect explanation, and emergency risk clarification. The LLMs are used in a controlled, prompt-engineered manner to ensure outputs remain educational, concise, and non-diagnostic.

- **Rule-Based Risk Scoring Engine**

Implements predefined safety rules to compute emergency risk scores based on symptom descriptions and medicine combinations. This deterministic layer ensures transparency, consistency, and explainability in risk assessment.

- **JSON & Structured Data Handling**

Enables standardized representation of extracted medicines, drug salts, side-effect logs, and AI outputs. Structured formats improve reliability, traceability, and downstream analysis.

- **Backend Logic (Core Processing Layer)**

Orchestrates communication between the Streamlit UI, OCR engine, medicine database, fuzzy matcher, rule-based risk system, and LLM services. This layer ensures modularity, scalability, and efficient execution across all MedSafe AI functionalities.

Component-Wise Architecture:

Component	Description
User Interface (Streamlit)	Interactive, tab-based dashboard that allows users to enter medicine names, upload prescription images, describe symptoms, log side effects, and view risk scores. Handles real-time updates, alerts, metrics, expanders, and session persistence for a smooth user experience.
Medicine Interaction Checker	Accepts user-entered medicine names and applies fuzzy matching to accurately identify medicines from the database. Cross-references known drug–drug interactions and generates clear safety warnings for detected risks.
Fuzzy Matching Engine (find_medicine)	Uses RapidFuzz string-matching algorithms to handle spelling variations, OCR noise, and partial inputs when mapping user-provided or extracted text to known medicine records.
Medicine Database (MED_DB)	A curated local database storing medicine names, metadata, and interaction warnings. Serves as the authoritative reference for interaction checks and safety rule evaluation.
Prescription OCR Module	Utilizes Tesseract OCR to extract raw text from uploaded prescription images. Acts as the first stage in converting unstructured prescription data into machine-readable text.
LLM-Based Medicine & Salt Extractor	Uses a large language model to parse OCR text and extract structured medicine names along with their active drug/salt components in strict JSON format for clarity and reliability.

Symptom & Doubt Solver	Accepts free-text symptom descriptions and provides basic rule-based advice. Enhances responses using LLM-generated educational guidance, including home remedies, lifestyle tips, breathing/yoga exercises, diet suggestions, and warning signs.
Side-Effect Monitor	Collects demographic information, medicines taken, dosage, and post-medication experiences. Generates concise, educational explanations highlighting possible contributing factors and one clear precaution to watch for.
Emergency Risk Predictor	Applies predefined safety rules to calculate a transparent emergency risk score based on symptoms and medicine combinations. Categorizes risk into clearly defined urgency levels with actionable guidance.
Risk Scoring Engine	Implements deterministic logic using symptom keywords and known dangerous drug combinations to compute normalized risk scores, ensuring explainability and consistency.
LLM Explanation Generator	Generates short, non-diagnostic explanations for interaction warnings, symptom guidance, side-effect analysis, and risk assessments, improving interpretability for users.
Session State Management	Maintains extracted data, side-effect logs, and analysis results within the Streamlit session to ensure continuity across user interactions.
JSON & Structured Data Handler	Ensures consistent storage and transfer of extracted medicines, salts, side-effect logs, and AI outputs using structured JSON formats.

Backend Logic Controller	Orchestrates communication between the UI, OCR engine, fuzzy matcher, medicine database, risk scoring logic, and LLM services. Ensures modular execution, scalability, and clean separation of concerns.
Helper Utilities	Includes utility functions for text cleaning, directory handling, time-stamping logs, device configuration, and error handling to support stable and maintainable execution.

Pre-requisites:

Before using **MedSafe AI**, ensure that your development environment is properly configured. The following components and tools are required to run the application smoothly and enable all AI-powered features.

1. Python Environment Setup

MedSafe AI is developed using **Python 3.10+**, leveraging libraries for web deployment, OCR processing, fuzzy matching, rule-based logic, and AI model interaction.

- Create and activate a dedicated virtual environment (e.g., `medsafe_env`) to isolate dependencies and maintain a clean setup.
- Official Python documentation: <https://www.python.org/downloads/>

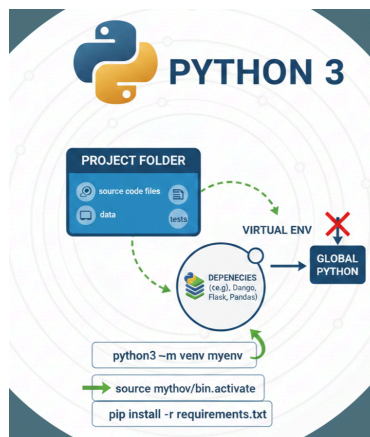


Figure 2: virtual environment workflow illustration

2. Streamlit Installation & Configuration

Streamlit powers the interactive front-end dashboard of MedSafe AI, enabling tab-based navigation, real-time analysis, and user-friendly visual outputs.

- Install Streamlit via pip:

pip install streamlit

- Streamlit documentation: <https://docs.streamlit.io/library/get-started/installation>
- Introductory tutorial: <https://www.youtube.com/watch?v=JwSS70SZdyM>

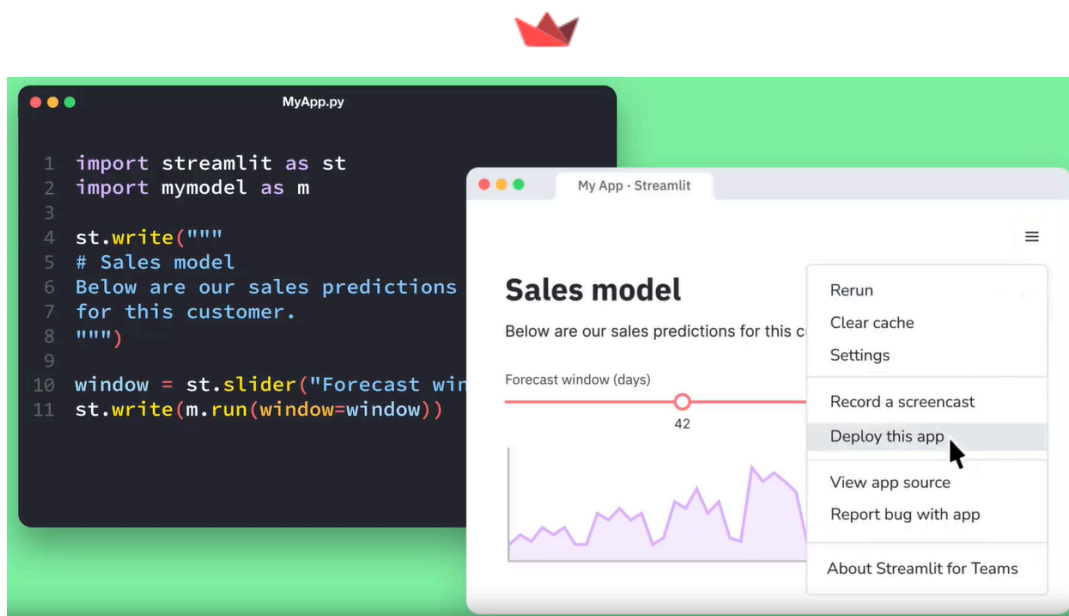


Figure 3: Streamlit UI

3. OCR Engine Setup (Tesseract OCR)

MedSafe AI uses **Tesseract OCR** to extract text from uploaded prescription images. Proper installation and configuration are required.

- Download and install Tesseract OCR for your operating system
- Official repository: <https://github.com/tesseract-ocr/tesseract>
- Ensure the Tesseract executable path is correctly configured in the code

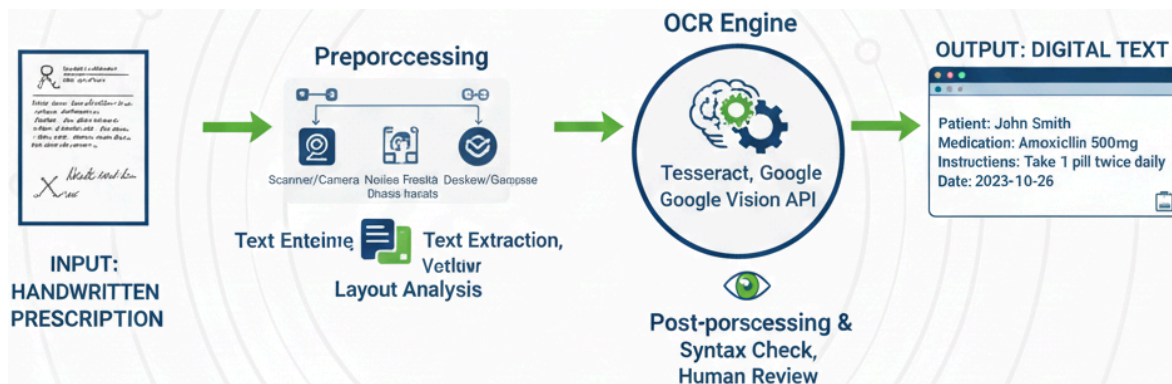


Figure 4: OCR Workflow

4. Library Installation & Core Dependencies

MedSafe AI relies on several Python libraries for text extraction, fuzzy matching, AI interaction, and UI rendering. All dependencies can be installed using a `requirements.txt` file.

Key libraries include:

- `streamlit` – Interactive web interface
- `pytesseract` – OCR text extraction
- `Pillow (PIL)` – Image loading and processing
- `rapidfuzz` – Fuzzy string matching for medicine detection
- `ollama` – Local LLM interaction (LLaMA 3)
- `json`, `datetime` – Structured data handling and logging

5. LLM Runtime Setup (Ollama)

MedSafe AI uses **LLaMA 3 via Ollama** for AI-powered summarization, extraction, and explanation tasks.

- Install Ollama on your system
- Pull the required model (e.g., `llama3`)
- Ollama documentation: <https://ollama.com>

6. Development Environment

A modern IDE is recommended for efficient development, testing, and debugging of the MedSafe AI codebase.

Recommended IDEs:

- Visual Studio Code: <https://code.visualstudio.com/>
- PyCharm (Community Edition): <https://www.jetbrains.com/pycharm/download/>

7. Optional Learning & Reference Resources

For deeper understanding and customization of MedSafe AI components:

- Streamlit Components Gallery: <https://streamlit.io/components>
- Tesseract OCR Documentation: <https://tesseract-ocr.github.io/>
- RapidFuzz Documentation: <https://maxbachmann.github.io/RapidFuzz/>
- Prompt Engineering for LLMs (General Reference): <https://www.promptingguide.ai/>

Project Flow:

1. Environment Setup and Dependency Configuration

This phase focuses on preparing a stable and isolated development environment for MedSafe AI and validating the integration of all foundational components.

- **Activity 1.1:** Create and activate a Python virtual environment (e.g., `medsafe_env`) and install all required dependencies, including `streamlit`, `pytesseract`, `Pillow`, `rapidfuzz`, `ollama`, and supporting standard libraries. This ensures dependency isolation and consistent execution across systems.
- **Activity 1.2:** Organize the project structure into modular components, such as medicine database management, OCR processing, fuzzy matching logic, AI prompt handlers, risk scoring utilities, and UI layout configuration. This modularization improves maintainability and scalability.
- **Activity 1.3:** Initialize the Streamlit application and validate seamless communication between the user interface and backend logic, ensuring that user inputs correctly trigger OCR, interaction checks, AI inference, and result rendering.

2. Core Logic Development (Medicine Safety & AI Reasoning Engine)

This phase implements the core intelligence behind MedSafe AI, combining deterministic safety rules with controlled generative AI outputs.

- **Activity 2.1:** Develop the **Medicine Interaction Checker**, enabling accurate medicine identification using fuzzy matching and cross-referencing with the curated medicine database to detect known interaction warnings.
- **Activity 2.2:** Implement the **Prescription OCR and AI Parsing Module**, which extracts raw text from prescription images and converts it into structured medicine and drug-salt information using large language models.
- **Activity 2.3:** Build the **Symptom Analysis, Side-Effect Monitoring, and Risk Scoring Engine**, integrating rule-based logic for emergency risk calculation with AI-generated educational explanations to maintain clarity and transparency.

3. Streamlit UI Implementation and User Interaction

This phase focuses on delivering an intuitive, interactive, and user-friendly healthcare safety interface.

- **Activity 3.1:** Design a multi-tab Streamlit layout covering medicine interaction checks, prescription OCR, symptom guidance, side-effect monitoring, and emergency risk prediction, with real-time updates and clear visual feedback.
- **Activity 3.2:** Configure robust user input handling for text, images, demographic details, and symptom descriptions, while managing session state to preserve results across interactions.
- **Activity 3.3:** Display structured outputs, AI-generated explanations, warnings, metrics, and expandable raw data sections in a clear and interpretable manner to support informed user understanding.

4. Testing, Optimization, and Deployment

This final phase ensures system reliability, performance, and readiness for real-world usage.

- **Activity 4.1:** Test all UI components and backend workflows, verifying accurate medicine detection, OCR extraction, AI responses, and risk scoring behavior across multiple scenarios.
- **Activity 4.2:** Validate safety logic, interaction detection accuracy, and consistency of AI-generated outputs to ensure educational reliability and non-diagnostic compliance.

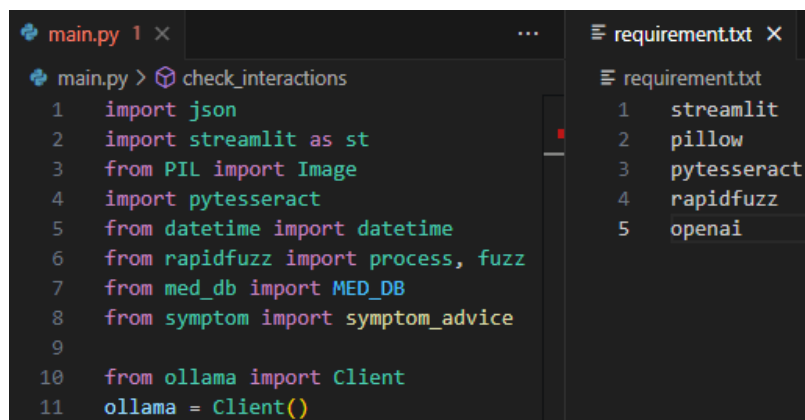
- **Activity 4.3:** Optimize overall performance, conduct end-to-end testing, and prepare the application for deployment in academic, research, or prototype healthcare settings.

MILESTONE 1: Environment Setup and Dependency Configuration

This foundational milestone establishes the technical environment required for building and deploying **MedSafe AI**. It ensures that all dependencies, frameworks, and system integrations are correctly configured to support medicine interaction analysis, prescription OCR, AI-driven reasoning, and interactive visualization workflows.

Activity 1.1: Python Environment and Dependency Installation

- Create and activate a dedicated virtual environment for MedSafe AI (e.g., `medsafe_env`) to ensure dependency isolation and reproducible execution.
- Install all required dependencies listed in `requirements.txt`, including:
`streamlit`, `pytesseract`, `Pillow`, `rapidfuzz`, `ollama`, and supporting Python libraries.
- Verify compatibility with **Python 3.10+** and confirm successful installation of all modules.
- Test library imports and ensure that core components—OCR processing, fuzzy matching, AI model interaction, and UI rendering—operate correctly without runtime errors.



```
main.py 1 x ... requirements.txt x
main.py > check_interactions
1 import json
2 import streamlit as st
3 from PIL import Image
4 import pytesseract
5 from datetime import datetime
6 from rapidfuzz import process, fuzz
7 from med_db import MED_DB
8 from symptom import symptom_advice
9
10 from ollama import Client
11 ollama = Client()

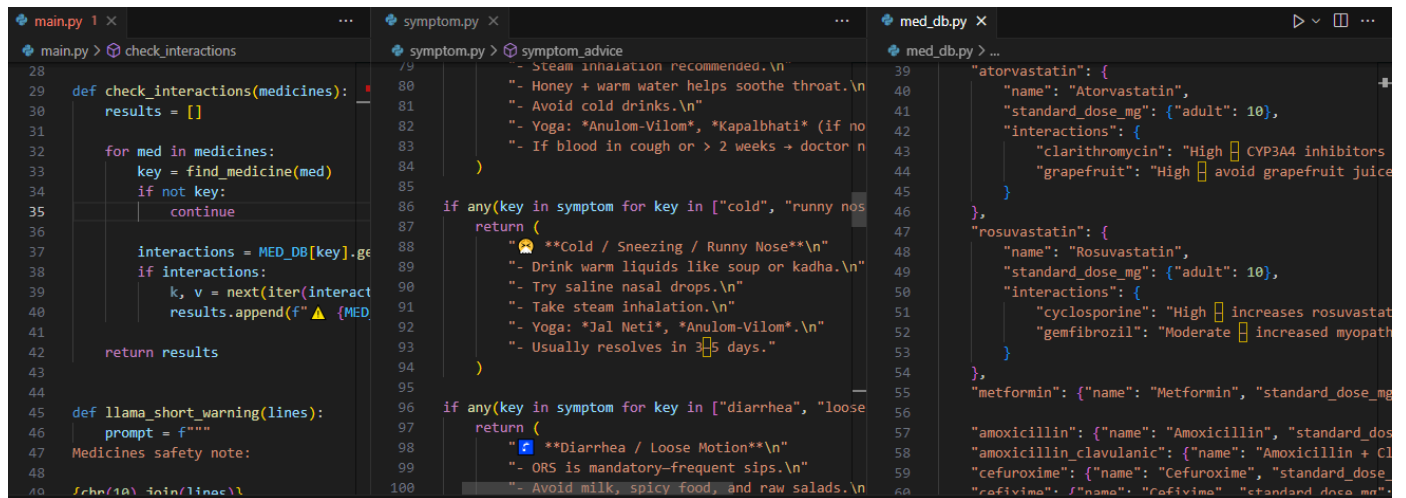
requirements.txt
1 streamlit
2 pillow
3 pytesseract
4 rapidfuzz
5 openai
```

Figure 5: Required & Imported Libraries

Activity 1.2: Project Structure Initialization

- Organize the project into a clear and maintainable structure using modular Python files to separate concerns and improve readability. Key components include:

- **streamlit_app.py** – Front-end interface and main application logic
 - **med_db.py** – Medicine database and interaction metadata
 - **symptom.py** – Rule-based symptom advice logic
 - **ocr_utils.py** – Prescription OCR and text extraction utilities
 - **risk_engine.py** – Emergency risk scoring and safety rules
- This modular organization ensures scalability, easier debugging, and cleaner integration between deterministic logic and AI-driven components.



```

main.py
28 def check_interactions(medicines):
29     results = []
30
31     for med in medicines:
32         key = find_medicine(med)
33         if not key:
34             continue
35
36         interactions = MED_DB[key].get('interactions', [])
37         if interactions:
38             k, v = next(iter(interactions.items()))
39             results.append(f"{med} {v}")
40
41     return results
42
43 def llama_short_warning(lines):
44     prompt = f"""
45     Medicines safety note:
46     {chr(10).join(lines)}
47
48     """
49
50     return prompt
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2
```

MILESTONE 2: Core Logic Development (Medicine Safety & AI Reasoning Engine)

This milestone focuses on building the core functional intelligence of **MedSafe AI**, responsible for analyzing medicine interactions, extracting structured information from prescriptions, interpreting symptoms, and generating safe, educational AI explanations. It establishes the essential backend logic required for intelligent, rule-aware, and AI-assisted healthcare safety workflows.

Activity 2.1: Medicine Interaction & Identification Module Development

- Implement the medicine identification logic using fuzzy string matching to accurately recognize medicine names from user input and OCR-extracted text, even in the presence of spelling variations or noise.
- Develop the interaction-checking workflow to cross-reference identified medicines against the curated medicine database and detect known drug–drug interaction warnings.
- Generate concise, AI-assisted safety summaries for detected interactions while ensuring non-diagnostic and educational output.

```
def extract_medicines_from_image(img):  
    text = pytesseract.image_to_string(img)  
    found = []  
  
    for word in text.split():  
        med = find_medicine(word)  
        if med:  
            found.append(med)  
  
    return list(set(found)), text
```

Figure 8: Logic Illustration

Activity 2.2: Prescription OCR and AI-Based Extraction Engine

- Integrate Tesseract OCR to extract raw textual data from uploaded prescription images.
- Implement LLM-driven parsing to identify medicines and their active drug or salt components, enforcing strict JSON output for structured and reliable data representation.
- Validate extracted data against the medicine database to improve accuracy and downstream safety analysis.

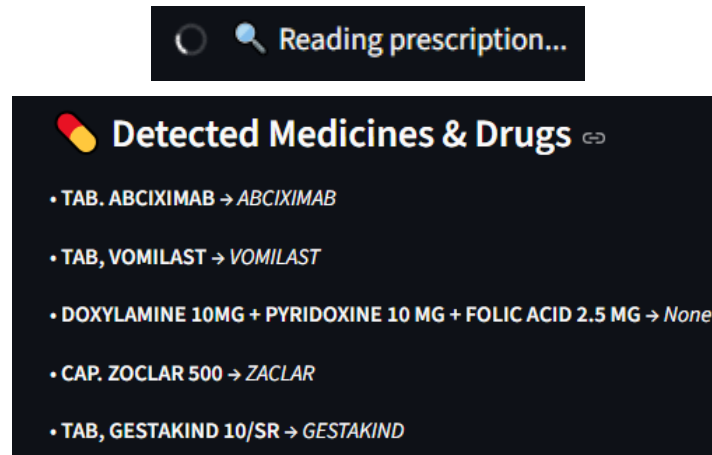


Figure 9: Detected prescription text Illustration

Activity 2.3: Symptom Interpretation, Side-Effect Analysis, and Risk Scoring Engine

- Build the symptom interpretation module to provide basic rule-based guidance, enhanced by AI-generated educational explanations including home remedies, lifestyle suggestions, and warning signs.
- Implement the side-effect monitoring logic to analyze user-reported experiences in the context of age, gender, medicines, and dosage.
- Develop the emergency risk scoring engine to compute transparent risk percentages based on predefined safety rules and known high-risk indicators.
- Integrate all core modules to ensure a smooth workflow from user input → analysis → AI explanation → UI presentation.
- Implement helper utilities for data handling, session management, and logging to maintain clean and traceable execution.

MILESTONE 3: Streamlit UI Implementation & User Interaction

This milestone establishes the interactive front-end layer of **MedSafe AI**, enabling users to access medicine safety tools, prescription analysis, symptom guidance, and risk assessment through a clean, responsive, and intuitive Streamlit dashboard. The focus is on usability, clarity, and safe presentation of AI-assisted healthcare insights.

Activity 3.1: User Interface and Multi-Tab Layout Design

- Design a structured multi-tab user interface to clearly separate core functionalities, including Medicine Interaction Checker, Prescription OCR, Symptom & Doubt Solver, Side-Effect Monitor, and Emergency Risk Predictor.
- Apply consistent styling and layout using Streamlit components to maintain professional appearance, readable typography, proper spacing, and clear visual hierarchy.
- Ensure responsiveness across different screen sizes and devices, enabling smooth interaction on desktops and tablets.

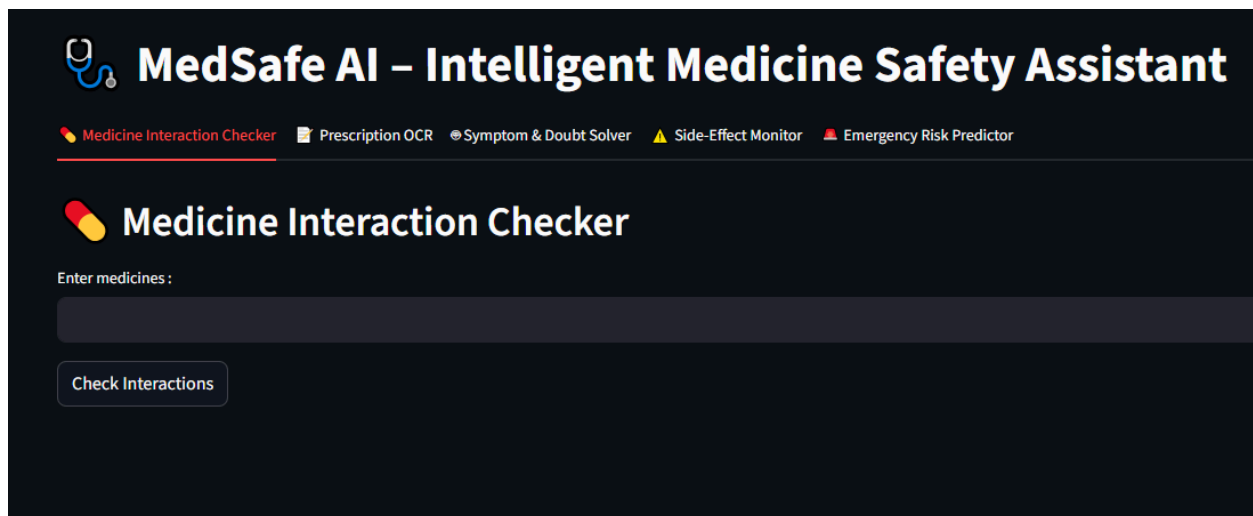


Figure 10: Dashboard Illustration

Activity 3.2: Input Configuration and Session State Management

- Implement interactive input components such as text fields, text areas, file uploaders, dropdowns, number inputs, and action buttons for capturing user data.
- Use Streamlit session state to persist extracted prescription data, interaction results, side-effect logs, and risk analysis outputs during tab navigation.
- Validate user inputs and implement basic error handling to manage missing, incomplete, or invalid entries gracefully.

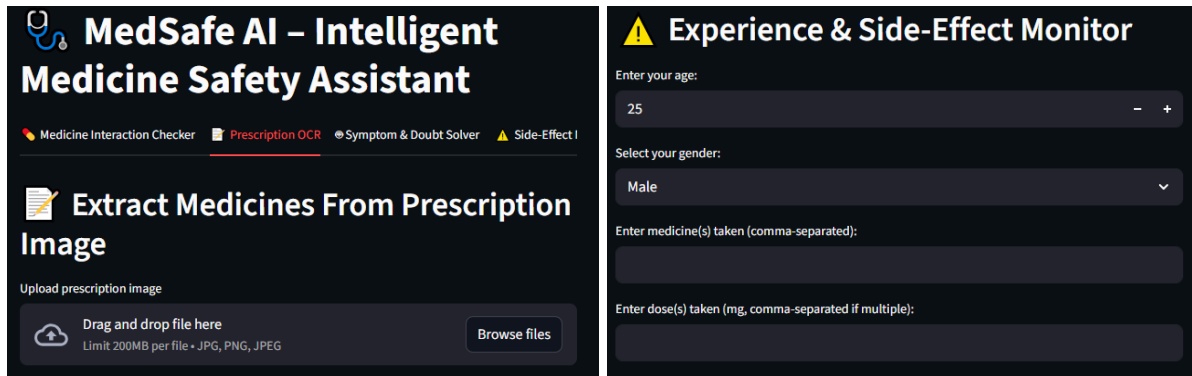


Figure 11: Interactive input components Illustration

Activity 3.3: Output Rendering and Data Visualization

- Display extracted medicines, interaction warnings, symptom guidance, and AI-generated explanations in a structured and readable format using alerts, metrics, and expandable sections.
- Render emergency risk scores using visual indicators and clearly labeled severity levels to support rapid understanding.
- Present raw OCR text and structured JSON outputs where appropriate to ensure transparency and traceability of AI-assisted results.
- Provide real-time feedback through color-coded messages (success, warning, error) to guide users safely through the analysis process.

🟢 AI Enhanced Advice: Here's a friendly medical guidance paragraph for you:

Hey there! We're here to help you tackle that pesky back pain and red eyes. Let's start with the back pain: 🚩 **Back Pain**. To ease the discomfort, try applying a hot compress to the affected area, avoiding long sitting hours, doing some gentle stretching, and incorporating yoga poses like *Bhujangasana*, *Cat-Cow*, and *Bridge pose*. If your pain radiates to your legs, it's possible that there's a nerve issue involved, so be sure to keep an eye out for that! Now, let's shift our focus to those red eyes: 🚩 **Eye Pain / Redness**. To soothe the discomfort, wash your eyes with clean cold water, avoid rubbing them (ouch!), reduce screen brightness, and try some yoga techniques like *Palming* and *Eye rotations*. If you notice any significant vision changes, please get an urgent evaluation.

In addition to these tips, here are a couple of extra home remedies you can try: applying a warm bath or shower to loosen up those muscles, and using a humidifier to add moisture to the air (this might help reduce eye irritation). To promote relaxation and reduce stress (which can exacerbate back pain), consider practicing some gentle breathing exercises like "box breathing" (inhale for 4 counts, hold for 4 counts, exhale for 4 counts, and hold again for 4 counts). Finally, make sure to fuel your body with a balanced diet that includes plenty of omega-3 fatty acids, vitamin D, and calcium-rich foods – these nutrients can help support overall health and well-being. And remember: if you experience any severe or persistent symptoms, be sure to consult with a healthcare professional.

Warning sign to watch: If you notice numbness, tingling, or weakness in your legs or feet, or if your back pain is accompanied by fever, chills, or difficulty urinating – these could indicate a more serious issue that requires medical attention.

Figure 12: AI Response to input medical symptoms

MILESTONE 4: Testing, Optimization, and Deployment

This milestone ensures that **MedSafe AI** meets performance, accuracy, reliability, and safety standards through structured testing, optimization, and validation. It prepares the platform for real-world usage and confirms that all healthcare safety workflows operate correctly in an end-to-end environment.

Activity 4.1: Functional Testing and Module Verification

- Test all user interface components, including text inputs, file uploads, buttons, tabs, and alerts, to verify correct behavior across all modules.
- Validate medicine interaction detection, prescription OCR extraction, symptom guidance responses, side-effect analysis, and emergency risk scoring for logical correctness and consistency.
- Ensure AI-generated outputs remain educational, non-diagnostic, and aligned with safety guidelines across multiple test scenarios.

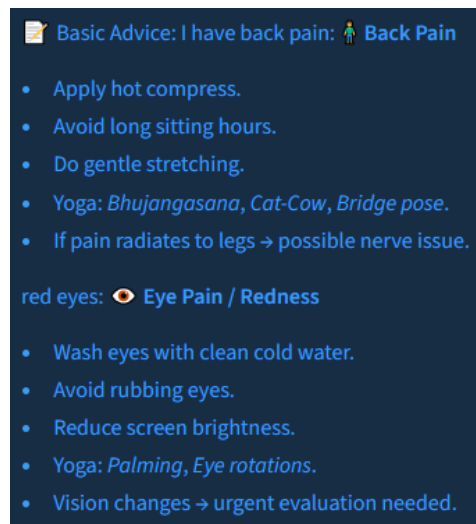


Figure 13: Testing on basic symptoms

Activity 4.2: Performance Testing and Optimization

- Measure application responsiveness and runtime performance for OCR processing, fuzzy matching, AI inference, and risk score calculation under varying input sizes.
- Optimize performance by minimizing redundant OCR calls, improving fuzzy matching efficiency, and managing AI model invocation frequency.

- Test system stability with multiple sequential requests, large symptom descriptions, and complex medicine combinations to ensure reliability under realistic usage conditions.

```
def find_medicine(name):  
    if not name:  
        return None  
  
    cleaned = name.lower().replace("+", " ").replace(",", " ").replace(".", " ").strip()  
    names = list(MED_DB.keys())  
    match, score, _ = process.extractOne(cleaned, names, scorer=fuzz.WRatio)  
    if score >= 80:  
        return match  
    return None
```

Figure 14: Logic behind OCR

Activity 4.3: Deployment Preparation and Final Validation

- Prepare the application for deployment on supported platforms such as local servers or Streamlit Cloud, ensuring environment variables and model dependencies are correctly configured.
- Conduct final end-to-end validation covering user input → analysis → AI explanation → UI output for all major workflows.
- Verify project structure, logging of side-effect experiences, error handling, and safe fallback mechanisms before release.

Conclusion:

MedSafe AI represents a meaningful advancement in AI-assisted healthcare safety and preventive awareness, transforming how users understand medicines, prescriptions, symptoms, and potential health risks through a single intelligent platform. By integrating OCR-based prescription analysis, fuzzy medicine identification, rule-based safety logic, and controlled large language model reasoning, the system delivers clear, structured, and educational insights that support safer decision-making without replacing professional medical consultation.

Built on an intuitive Streamlit interface, MedSafe AI provides a seamless and accessible user experience. Users can easily check medicine interactions, extract drug information from prescription images, describe symptoms, monitor side effects, and assess emergency risk levels—all within a clean, responsive dashboard. The combination of deterministic safety rules

and AI-generated explanations ensures transparency, consistency, and interpretability across all outputs.

Looking ahead, MedSafe AI holds strong potential for future expansion. Enhancements such as a larger medicine knowledge base, personalized risk profiling, multilingual support, and deeper integration with wearable or health-monitoring systems could further strengthen its impact. Ultimately, this project demonstrates how responsibly designed AI systems can enhance health literacy, promote early risk awareness, and empower individuals with reliable, user-friendly tools for medicine safety and preventive healthcare education.