SmartSDLC-AI-EnhancedSoftware DevelopmentLifecycle

ProjectDocumentation

1. Introduction

- Projecttitle:SmartSDLC-AI-EnhancedSoftwareDevelopmentLifecycle

- Teammember: Dhivya Dharshini G
- Teammember: Divya Dharshini S
- Teammember: Monish S
- Teammember: Infant Robin J

2. projectoverview

- Purpose:

ThepurposeofaSustainableSmartCityAssistantistoempowercitiesandtheir residentstothriveinamoreeco-consciousandconnectedurbanenvironment.By leveragingAIandreal-timedata,theassistanthelpsoptimizeessentialresourceslike energy,water,andwaste,whilealsoguidingsustainablebehaviorsamongcitizensthrough personalizedtipsandservices.Forcityofficials,itservesasadecisionmaking

partner—offering clearinsights, forecasting tools,andsummarizations ofcomplexpoliciestosupportstrategicplanning.Ultimately,thisassistantbridges technology,governance,andcommunityengagementtofostergreenercitiesthataremore efficient,inclusive,andresilient.

- Features:

ConversationalInterface

KeyPoint:Naturallanguageinteraction

Functionality:Allowscitizensandofficialstoaskquestions,getupdates,and receive guidance in plain language

PolicySummarization

KeyPoint:Simplifiedpolicyunderstanding

Functionality:Convertslengthygovernmentdocumentsintoconcise,actionable summaries.

ResourceForecasting

KeyPoint:Predictiveanalytics

Functionality:Estimatesfutureenergy,water,andwasteusageusinghistoricalandreal-timedata.

Eco-TipGenerator

KeyPoint:Personalizedsustainabilityadvice

Functionality:Recommendsdailyactionstoreduceenvironmentalimpactbasedonuser behavior.

CitizenFeedbackLoop

KeyPoint:Communityengagement

Functionality:Collectsandanalyzespublicinputtoinformcityplanningandservice improvements.

KPIForecasting

KeyPoint:Strategicplanningsupport

Functionality:Projectskeyperformanceindicatorstohelpofficialstrackprogressandplan ahead.

AnomalyDetection

KeyPoint:Earlywarningsystem

Functionality:Identifiesunusualpatternsinsensororusagedatatoflagpotentialissues.

MultimodalInputSupport

KeyPoint:Flexibledatahandling

Functionality:Acceptstext,PDFs,andCSVsfordocumentanalysisandforecasting.

StreamlitorGradioUI

KeyPoint:User-friendlyinterface

Functionality:Providesanintuitivedashboardforbothcitizensandcityofficialstointeract with the assistant.

3. Architecture

Frontend(Streamlit):

ThefrontendisbuiltwithStreamlit,offeringaninteractivewebUIwithmultiplepages includingdashboards,fileuploads,chatinterface,feedbackforms,andreportviewers. Navigationishandledthroughasidebarusingthestreamlit-option-menulibrary.Each pageismodularizedforscalability.

Backend(FastAPI):

FastAPIservesasthebackendRESTframeworkthatpowersAPIendpointsfordocument processing,chatinteractions,ecotipgeneration,reportcreation,andvectorembedding.It isoptimizedforasynchronousperformanceandeasySwaggerintegration.

LLMIntegration(IBMWatsonxGranite):

GraniteLLMmodelsfromIBMWatsonxareusedfornaturallanguageunderstandingand generation.Promptsarecarefullydesignedtogeneratesummaries,sustainabilitytips,and reports.

VectorSearch(Pinecone):

UploadedpolicydocumentsareembeddedusingSentenceTransformersandstoredin Pinecone.Semanticsearchisimplementedusingcosinesimilaritytoallowuserstosearch documentsusingnaturallanguagequeries.

MLModules(ForecastingandAnomalyDetection):

LightweightMLmodelsareusedforforecastingandanomalydetectionusingScikit-learn. Time-seriesdataisparsed,modeled,andvisualizedusingpandasandmatplotlib.4.Setup Instructions

Prerequisites:

- Python3.9orlateropipandvirtualenvironmenttoolsoAPIkeysforIBMWatsonx andPinecone oInternetaccesstoaccesscloudservices
-

InstallationProcess:

- ClonetherepositoryoInstalldependenciesfromrequirements.txtoCreatea.env fileandconfigurecredentialsoRunthebackendserverusingFastAPIoLaunchthe frontendviaStreamlitoUploaddataandinteractwiththemodules
-

5. FolderStructure

App/–ContainsallFastAPIbackendlogicincludingrouters,models,andintegration modules.

App/api/–SubdirectoryformodularAPIrouteslikechat,feedback,report,anddocument vectorization.

Ui/–ContainsfrontendcomponentsforStreamlitpages,cardlayouts,andformUIs.

Smart_dashboard.py–EntryscriptforlaunchingthemainStreamlitdashboard.

Granite_llm.py–HandlesallcommunicationwithIBMWatsonxGranitemodelincluding summarization and chat.

Document_embedder.py–ConvertsdocumentstoembeddingsandstoresinPinecone.

Kpi_file_forecaster.py–Forecastsfutureenergy/watertrendsusingregression. Anomaly_file_checker.py –FlagsunusualvaluesinuploadedKPIdata.Report_generator.py – ConstructsAI-generatedsustainabilityreports.

6. RunningtheApplication

Tostarttheproject:

➢ LaunchtheFastAPIservertoexposebackendendpoints.

➢ RuntheStreamlitdashboardtoaccessthewebinterface.

➢ Navigatethroughpagesviathesidebar.

➢ UploaddocumentsorCSVs,interactwiththechatassistant,andview

Outputslikereports,summaries,andpredictions.

➢ Allinteractionsarereal-timeandusebackendAPIstodynamicallyupdatethe frontend.

Frontend(Streamlit):

ThefrontendisbuiltwithStreamlit,offeringaninteractivewebUIwithmultiplepages includingdashboards,fileuploads,chatinterface,feedbackforms,andreportviewers. Navigationishandledthroughasidebarusingthestreamlit-option-menulibrary.Each pageismodularizedforscalability.

Backend(FastAPI):

FastAPIservesasthebackendRESTframeworkthatpowersAPIendpointsfordocument processing,chatinteractions,ecotipgeneration,reportcreation,andvectorembedding.It isoptimizedforasynchronousperformanceandeasySwaggerintegration.

7. APIDocumentation

BackendAPIsavailableinclude:

POST/chat/ask–AcceptsauserqueryandrespondswithanAI-generatedmessage

POST/upload-doc–UploadsandembedsdocumentsinPinecone

GET/search-docs–Returnssemanticallysimilarpoliciestotheinputquery

GET/get-eco-tips–Providessustainabilitytipsforselectedtopicslikeenergy,water,or waste

POST/submit-feedback–Storescitizenfeedbackforlaterrevieworanalytics

EachendpointistestedanddocumentedinSwaggerUIforquickinspectionandtrial duringdevelopment.

8. Authentication

EachendpointistestedanddocumentedinSwaggerUIforquickinspectionandtrial duringdevelopment.

Thisversionoftheprojectrunsinanopenenvironmentfordemonstration.

However,securedeploymentscanintegrate:

- Token-basedauthentication(JWTorAPIkeys)
- OAuth2withIBMCloudcredentials
- Role-basedaccess(admin,citizen,researcher)
- Plannedenhancementsincludeusersessionsandhistorytracking.8. Authentication

9. UserInterface

Theinterfaceisminimalistandfunctional,focusingonaccessibilityfornontechnicalusers. Itincludes:

Sidebarwithnavigation

KPIvisualizationswithsummarycards

Tabbedlayoutsforchat,ecotips,andforecasting

Real-timeformhandling

PDFreportdownloadcapability

Thedesignprioritizesclarity,speed,anduserguidancewithhelptextsand intuitive flows.

10. Testing

Testingwasdoneinmultiplephases:

UnitTesting:Forpromptengineeringfunctionsandutilityscripts

APITesting:ViaSwaggerUI,Postman,andtestscripts

ManualTesting:Forfileuploads,chatresponses,andoutputconsistency

EdgeCaseHandling:Malformedinputs,largefiles,invalidAPIkeys

EachfunctionwasvalidatedtoensurereliabilityinbothofflineandAPIconnectedmodes.

11. screenshots

12. KnownIssues

13. Futureenhancement