Date:

IV SEMESTER – MCA491P Major Project

QueryForge

Synopsis

Introduction
This project integrates two advanced AI tools: a Code Generator & Debugge**r** and a Retrieval-Augmented QA Bot. The domain spans AI-assisted software development and intelligent document retrieval, which are increasingly crucial in boosting developer productivity and automating information access. Existing models like CodeT5, StarCoder, and GPT-3.5/4 have shown promise in code generation and natural language understanding, but often lack domain-specific adaptability and effective debugging. Similarly, traditional QA systems do not provide context-aware, document-specific answers—especially in multilingual scenarios. This project addresses these gaps by combining code intelligence and semantic search.

Methodology
The system uses pre-trained LLMs for code tasks and RAG (Retrieval-Augmented Generation) for document QA. The methodology includes collecting open-source code datasets, fine-tuning models, implementing a web interface, and integrating vector search (via FAISS/Chroma) for document chunk retrieval. Technologies used include Python, Hugging Face Transformers, LangChain, OpenAI API, PyMuPDF, Sentence Transformers, and React or Flask for the frontend. Modules include: dataset collection, LLM integration, code generation/debugging, document upload & parsing, chunk embedding, vector storage, and multilingual QA.

ExpectedOutcome
The expected outcomes include a working full-stack application capable of generating and debugging code from problem statements, and answering any question from uploaded PDFs in English or Manipuri. The system will display retrieved source paragraphs as references for answers. With integrated LLMs, we aim for an efficiency of over 90% in answer relevance and high success rates in code generation/debugging tasks**,** making this a robust productivity and learning tool.

 Kh Dipayan Singha(1RV23MC045)   **Dr Usha J**