Adam Savinkin

AI Engineer | Full Stack Developer

adam.savinkin9@gmail.com +380509264684, +447414076728 Ukraine
AdamSavinkin adam-savinkin @Pro_React_777

Summary

AI-focused Full Stack Developer with 7+ years of experience delivering high-performance web applications and AI-powered solutions. Specialized in building advanced AI systems, including RAG-based chatbots and intelligent platforms. Recognized for exceptional expertise in optimizing project architecture, improving code quality, and enhancing application speed and scalability across both front-end and back-end development using RESTful API principles. Strong track record in performance tuning and ensuring long-term maintainability for complex software systems.

Education

Ivan Franko National University of Lviv Sep 2014 - Jan 2019

Master's degree, Computer Software Engineering

Skills and Tools

- Back-End: Node.js, Express.js, Django
- Front-End: React.js, Vue.js, Next.js, Nuxt.js
- Database: MySQL, PostgreSQL, MongoDB, Supabase
- API: Restful API, postman
- Version Control & Deployment: Git, Docker
- Cloud & Hosting: AWS, EC2 Instances
- AI & Automation: OpenAI, ChatGPT, LLM, LangChain

Language

- English: Fluent
- Ukrainian: Native or Bilingual

Professional Experience

- Software Engineer | MindfulProd (February 2020 - March 2021)

During my tenure at MindfulProd, I served as a software engineer and made significant contributions to the Telerad project. Leveraging my expertise in Javascript, Python, React.Js. I birdged the reporting gap between radiologists and doctors. I created a structure that enabled doctors to access patient results as DICOM files, write studies on the reports, and create studies for further analysis.

- Full Stack Developer | WamiSoftware (April 2021 - May 2022)

During my time at WamiSoftware, I served as a Full Stack Developer and made significant contributions to the Asiansocial project. Leveraging my expertise in Javascript, Java, SprintBoot, React.js, and AWS.

I lead the development of frontend and backend and delivered successfully.

- AI Engineer | TheGridBiz (July 2022 - January 2024)

During my time at TheGridBiz, Leveraging my expertise in Generative AI, National Langue Process, Large Language Model and Vector Database. I optimized the accuracy of suggestions and analyzed results which will upload some social media. I also built a responsive and user-friendly Frontend that enabled users to access and interact easily with some companie's brands and services.

In particular, the following projects: https://replient.ai

My role: AI Engineer and Backend developer

This project is for agencies automate comment management, maintain brand consistency, and increase engament - without the hassle on social platform such as Facebook, TikTok, Instagram, LinkedIn, etc.

1) For RAG system+, I utilized the Pinecone as Vector Database to retrive the relevant sections based on the data or information. I can use one according to the requirements of the project.

In my project, I used the Pinecone and Supabase extension (Supabase provides extension to store embedded data in field in the table). The reason that I used the supabase is that it can be used for Backend,

Authentification, and Vector DB.

- I implemented the features that get extract text and structure from the files such as Pdf, Docx, Excel, Html, and scraped websites. (To scrape the websites, I got the page lists from Robots.txt typically or used the Scrapy API when I can't scrape the websites).

- I implemented a chunking process, which is a critical component of RAG system where large datasets are divided into smaller, manageable pieces known as "chunks".

- I implemented the embedding process by converting each chunk info embeddings, using the OpenAI model(text-embedding-ada-002).

- I designed the vector database. And I saved the chunks and their corresponding embedded data using a key(embedded data converted)-value(each chunk) strcture or other schemas.

2) User process.

In my project, I gathered user comments from social platforms such as Facebook, LinkedIn, TikTok, and Instagram through the webhook API that Facebook provides.

- I converted the user's comments into embedded data to perform semantic searches on the vector database based on the uploaded data or information.

- After conducting the semantic search, I extracted the relevant sections(the divided chunks).

3) Prompt Design

I believe it is crucial to formulate effective prompts to obtain the desired results from LLM.(OpenAI, Gemini, Perplexity, depending on your preference).

- Base prompt: This is the prompt designed to eclicit the desired results from the provided data and user's input

- Context: This prompt combines the extracted chunks to provide necessary information.

- User prompt: This is the prompt that user inputs (in my project, it is user's comments).

- Final prompt: Combined with Base prompt, Context, User prompt, and others. (In my project, I used GPT-3.5-turbo, GPT-4 models)

4) Subscription:

I have integrated Stripe to handle subscription management.

- As a reference, I’d like to mention one thing.

I consider it a principle to ensure project structure optimization, code refinement, performance improvement, and maintainability when developing software.

This is a typical example:

Legacy Web Application Refactoring & Optimization
Stack: Express.js, React.js, MongoDB

Tasked with a version upgrade and performance improvement of an existing full-stack web application built using Express.js, React.js, and MongoDB. Upon detailed analysis, I identified critical performance bottlenecks across both back-end and front-end layers:

- Back-end issues:
  - Inefficient database schema and unnecessary, repetitive database calls
  - Lack of modularization in the core engine and middleware logic
- Front-end issues:
  - Incorrect implementation of Redux architecture, including improper use of store and failure to utilize Redux-based communication modules for API integration
  - Fundamental misapplication of Redux-Saga and Redux-Thunk, leading to unmanageable async logic and data flow
  - Misuse of React Function Component lifecycles, causing persistent warnings and runtime errors

I restructured the project architecture, modularized the codebase, implemented proper async control using Redux middleware, and enforced clean lifecycle management in React components. These optimizations significantly improved application speed, reduced bugs, and ensured long-term maintainability.

One major concern I observed during the project was the inconsistent use of JavaScript syntax. Some developers failed to properly apply ES6/ES7/ES12 features, resulting in mixed coding styles within single component files. This lack of consistency significantly degraded code quality and maintainability.

I will always be responsible for my results and strive to provide the best satisfaction to you.

My Life Philosophy

I believe consistent learning and sincere effort are the most honest paths to success. I realize my value through acquiring new skills and solving real-world problems.

All The Best

Adam Savinkin