AI FOR SOFTWARE ENGINEERING - ISIS4226

BETSMART - DEVELOPMENT REPORT

Group #2

Harold Nicolás Coca Peña

Germán Alberto Rojas Cetina

Daniel Felipe Acevedo

Luimarco Carrascal Diaz

GitHub Repository: https://github.com/NicolasCoca06/PROYECTO-FINAL.git

Description:

This report documents the development and implementation of BetSmart, an online sports betting platform designed as part of the ISIS 4226: AI for Software Engineering Project course. The primary goal of this project was to apply the concepts and methodologies learned in the course to build a fully functional software application using AI tools. BetSmart allows users to register, view sports event listings, manage betting odds, place bets, and track live scores, providing a seamless and engaging user experience.

The project aimed to provide hands-on experience in full-stack development, emphasizing both front-end and back-end development. Additionally, it focused on integrating artificial intelligence into the software engineering process, enhancing the platform's functionality with AI-driven insights and recommendations.

This report includes detailed documentation of the system architecture, design decisions, implementation process, and the challenges encountered. It also highlights the contributions of each team member and reflects on the lessons learned from using AI tools in the development process. By hosting all artifacts related to the project in a GitHub repository, the report ensures transparency and enables evaluation of each member's contributions.

Feature Distribution:

Feature	People in charge
User authentication and Account Managment	Harold Nicolás Coca Peña
Event Listing and Filtering	German Alberto Rojas
Odds Management	Luimarco Carrascal
Bet Placement and Transaction Proccesing	Daniel Acevedo
Live Scores and Results	Germán Alberto Rojas, Luimarco Carrascal
Insight recommendation	Daniel Acevedo
Project documentation	Harold Nicolás Coca Peña

Harold Nicolás Coca Peña:

- User Authentication and Account Managment.

- Odd Management.
- Bet Placement.
- Bet Confirmation and History.
- Result Notifications.

Germán Alberto Rojas Cetina:

- Odd Management.
- Event Listings.
- Live Scores.
- Event Filtering.

Daniel Felipe Acevedo:

- Gemini AI service module
- Recommender bets system
- Bet analysis service (User betting history and analyzer of betting patterns)

Luimarco Carrascal Diaz

- Odds Management and Events Listing
- Repository Organization
- Live Scores and Results
- Lessons Learned

Reflection about non-AI related challenges and limitations encountered:

One of the significant limitations we encountered during our software development project was that the API we utilized, in its free version, only allowed 100 requests per day. This constraint significantly impacted on our ability to test the code thoroughly. Once we exhausted the daily limit, we were unable to continue testing, leading to delays in identifying and fixing bugs, verifying the functionality of new features, and ensuring the overall reliability of the application. This limitation hindered the iterative development process, as we had to wait until the next day to resume testing, thus affecting our productivity and timelines.

Additionally, we faced the challenge of not having full proficiency in some of the tools used in our tech stack. This lack of expertise slowed down the development process as we had to invest extra time in learning and troubleshooting. It also increased the likelihood of encountering implementation issues and required us to seek external help or resources to bridge our knowledge gaps. This limitation emphasized the importance of having a well-rounded skill set within the team or ensuring access to adequate training and support when adopting new technologies.

Personal reflection on the use of AI tools

Harold Nicolás Coca Peña:

Personally, as a team member, I used GitHub Copilot, which helped streamline the development process and resolve some bugs more quickly. Additionally, I utilized ChatGPT to consult the documentation for tools that I wasn't very familiar with.

Challenges:

One of the main challenges was that AI tools like GitHub Copilot and ChatGPT don't have the complete context of the project. This means that the suggestions they provide might not always align perfectly with our specific requirements. Therefore, it's crucial to be very careful when implementing AI-generated suggestions to ensure they fit well within the project's overall architecture and objectives.

Lessons Learned:

A valuable lesson I learned from integrating these AI tools into our workflow is the importance of maintaining a critical eye. While AI can significantly enhance productivity and problem-solving, it's essential to double-check and validate AI-generated solutions against our project's context and goals. This ensures that we harness the benefits of AI without compromising the integrity and coherence of our codebase.

Germán Alberto Rojas Cetina

Throughout this project’s development, I leveraged several AI to enhance my productivity and solve problems more efficiently. Specifically, I used GitHub Copilot to streamline code writing, ChatGPT for documentation queries and clarifying unfamiliar technologies, and Claude AI for detailed explanations and assistance with complex problem-solving.

Challenges:

One of the main challenges was effectively integrating these AI tools into our workflow. While incredibly useful, each has its own strengths and limitations. For instance, GitHub Copilot excels at suggesting code, but its suggestions sometimes misalign with our coding conventions. ChatGPT is versatile but occasionally provides outdated information. Claude AI offers detailed explanations but can be overly verbose for tasks requiring quick responses.

Lessons Learned:

Working with these AI tools taught me the importance of balancing the efficiency they offer with the need to understand and validate their outputs. I learned to utilize each tool for its specific strengths: Copilot for rapid code generation, ChatGPT for general queries, and Claude for deeper analysis.

I also realized the importance of providing clear and precise context to these AIs to obtain more relevant results. Moreover, I developed the habit of always verifying AI-generated information and code before integrating it into our project, ensuring it aligns with our specific standards and requirements.

The integration of AI tools proved to be a valuable learning process that significantly improved our efficiency, provided they are used critically and consciously. It highlighted the importance of maintaining human oversight and combining AI capabilities with our expertise to achieve optimal results in software development.

Daniel Felipe Acevedo

Through my development I used different AI tools that helped me to be more productive and saved me a lot of time. I used Codium to complete my code and Chat-GPT 4o to give me some guides on parts where I was a little lost and needed some assistance.

Challenges:

Clearly, the main problem of using AI to code is that it can have many mistakes and the developer needs to fit the solution given from the AI to the specific project that the developer is working on. Sometimes GPT gave me outdated solutions or solutions that could harm other code of the project. Codium sometimes did not predict too well was I was intended to do and skipped some important blocks of code.

Lessons learned:

Using these AI tools allowed me to understand some concepts that I did not have very clear as the project assumed that every student had experience developing frontend and backend so using these tools taught me efficiently how to use different technologies required for this project. On the other hand, it made me realize that AI is not perfect and doesn't have the answer for everything. As a developer you need to know what is happening in the code so if a problem occurs you can fix it more easily. In conclusion, using AI tools and knowing how to use them is going to be essential in the future as a developer or engineer but still needs to understand the logic and face major problems.

Luimarco Carrascal Diaz

I utilized tools like claude sonnet 3.5 by anthropic, the gemini API for recommendation systems and chat gpt 4.o for generating the structure of the code, other non AI tools were node js, the sports api, vs code, mongo db, github, git, and windows cli, I found the AI tools useful because I was faster in the implementation of my features and that helped me in productivity

Lessons Learned:

Project management is a difficult thing, specially the coordination among so much member involved in the work, I also learned that communication is important, and respecting the deadlines too, so talking more would be fundamental for me in my professional life, and the tools are just so useful if you know how to use them, also i have to read a lot of documentation to understand the full scope of everything available in the marketplace and i find it fundamental to talk with the teachers more in order to keep them informed about everything that is going on.

Challenges:

I find it challenging to apply what is in the tutorials to a real project, because there are real needs which some of them are required and some of them are not so I have to filter the information in order to present what is just useful for the project, and also the setting up was a bit difficult for me because this is a new computer my Hewlett Packard, so I had to do everything from scratch, that was time-consuming and this Rythm of work is exhausting as well every day four hours working plus the time needed at home to work more, I had to adapt to all of these things which was challenging but it is a taste of what the real world is and I want to improve in that daily life style as well.