5-1 Journal Entry: Computer Science Trends & Artifact Update

Noah Khomer

Southern New Hampshire University

CS-499 (Capstone Project)

Joseph Martinez

10/5/2025

5-1 Journal Entry: Computer Science Trends & Artifact Update

Computer science is starting to become a long history. It comes down to innovation and discovery in every field that connects everyone in the world. For this journal entry, I will be discussing two emerging trends in computer science. One of the trends that has become a huge thing is generative AI and language models.

Generative AI models like ChatGPT-5 and Photopea are starting to transform what is possible in both the creative world and the technical world. The thing about these systems is that they can be used for a variety of different things. By establishing an API key, for example, these systems can write full code, generate marketing copy, synthesize research, and even assist in decision-making across different industries.

The trend is now starting to change how we will approach software development soon, with AI-assisted programming that is speeding up the development lifecycle by recommending different algorithms to use based on the code you already have. It improves debugging for low-code platforms. It is also starting to prompt new fields of study around prompt engineering and the ethical use of AI.

For example, we all know that AI can have a lot of bias when it comes to the training model because it represents human stereotypes. Consumers are starting to benefit a lot from this personalized experience that is powered by AI. For example, chatbots, AI tutors, and even content generation for platforms like YouTube. As someone enthusiastic about both AI and full-stack development, I am very aligned with this trend.

I am starting to understand reinforcement learning agents and integrating AI APIs into the web. Applications can improve how the application itself is built. However, at the same time, there are so many ethical concerns regarding that software that I only use it as a reference and never for full generation. A lot of these codes are starting to generate bad content, and security is an important thing when it comes to AI full-stack development.

Another huge trend that has come to the computer science field is WebAssembly. The concept of WebAssembly is amazingly simple. What it is trying to achieve is native performance inside browsers. So, for example, languages like C, Rust, and Go can be used in web applications, which can drastically improve what is possible on the front-end side. It is redefining the boundary between desktop and web apps. For example, we see full CAD software, machine learning, and even gaming engines running entirely in the browser without any use of plugins or hardware processing.

There is a lot of impact on consumers. For example, consumers can download games and experience them faster. A lot more powerful web apps can run like native software than operating systems. WebAssembly can break the language barrier and open doors to build cross-platform apps. So, this is especially important because there are distinct types of operating systems, such as Linux, Windows, and macOS, and not all of them work at the same level.

However, with WebAssembly, we can have those apps inside the browser, so we can experience them regardless of what we use. I am very enthusiastic about front-end-heavy app development because my whole career has been nothing but web development.

Therefore, one of the things I have learned is that the psychological point of user design is always focused on performance Therefore, learning more about the WebAssembly language is something that I am very curious about so it can future-proof my skill set, especially for browser applications.

| **Checkpoint** | **Software Design and Engineering** | **Algorithms and Data Structures** | **Databases** |
| --- | --- | --- | --- |
| **Name of Artifact Used** | Pirate Agent (CS-370) | Pathfinding Agent (CS-370) | Travlr Getaways (CS-465 → CS-499) |
| **Status of Initial Enhancement** | Improved UI and architecture | Enhanced logic and algorithm structure | Refactored schema and relationships |
| **Submission Status** | Submitted | Submitted | Submitted |
| **Status of Final Enhancement** | Completed | Completed | Completed |
| **Uploaded to ePortfolio** | Yes | Yes | Yes |
| **Status of Finalized ePortfolio** | Finalized and submitted | Finalized and submitted | Finalized and submitted |

References

Feng, J. (2023). *Chart Understanding with Large Language Model*. https://doi.org/10.31224/3401

*Free online photo editor*. Photopea. (n.d.-a). [https://www.photopea.com/](https://www.photopea.com/%20)

MozDevNet. (n.d.). *Webassembly*. MDN Blog RSS.

<https://developer.mozilla.org/en-US/docs/WebAssembly/Guides/Concepts>

Photopea. (n.d.-b). *Photopea/photopea: Photopea is online image editor*.

<https://github.com/photopea/photopea>

*Webassembly*. WebAssembly. (n.d.). [https://webassembly.org/](https://webassembly.org/%20)