**5-1 Journal: Computer Science Trends and Artifact Update**

**5-1 Journal: Computer Science Trends and Artifact Update**

Part One: Emerging Trends in Computer Science

Technology keeps changing fast, and there are always new trends shaping the future. Two big trends that are making an impact right now are Artificial Intelligence (AI) advancements and serverless computing.

1. Artificial Intelligence (AI) Advancements

* Significance: AI is getting smarter and more common in everyday life. From self-driving cars to AI chatbots like ChatGPT, AI is making big improvements in how we interact with technology. Companies are using AI to automate tasks, analyze data, and even generate code.
* How It Will Change Computer Science: AI is changing the way software is built. It helps developers write and debug code faster and improves cybersecurity by detecting threats automatically. AI is also being used to create better search engines, recommendation systems, and automation tools.
* How It Will Change People’s Experience: AI will continue making life easier. Customer service bots will become more human-like, and AI assistants will help with daily tasks like scheduling meetings or even writing emails. Workers may also need to learn how to work alongside AI instead of being replaced by it.
* How It Fits My Career Interests: Since I want to be a full-stack developer, AI can help improve the way I build apps. I might use AI to create chatbots for websites or recommendation systems for e-commerce platforms. Knowing how AI works will make me more valuable in the job market.

**2. Serverless Computing**

* Significance: Serverless computing is a way to run applications without worrying about managing physical servers. Cloud platforms like AWS, Google Cloud, and Azure handle the server maintenance, so developers can focus just on coding.
* How It Will Change Computer Science: It makes web development faster and more efficient. Developers don’t need to manage infrastructure, which saves time and money. It also allows applications to scale easily when more users start using them.
* How It Will Change People’s Experience: Websites and apps will load faster, crash less often, and be cheaper to run. Smaller businesses can compete with larger companies because they don’t need expensive server setups.
* How It Fits My Career Interests: Since I want to be a full-stack developer, learning about serverless computing will help me build better apps. I plan to look into AWS Lambda and Firebase Functions for future projects to make deployment easier.

**Which Course Outcomes Have I Achieved?**

Achieved so far:

* Improved React project structure by organizing components better.
* Optimized the shuffle algorithm using Fisher-Yates.
* Implemented a queue system to handle the deck more efficiently.
* Added tie-breaker mechanics so the game works more like real War.

**Still working on:**

* Connecting the database for storing game stats.
* Improving UI/UX to make the game look better.
* Finalizing my ePortfolio with descriptions and project examples.

Part Two:

|  |  |  |  |
| --- | --- | --- | --- |
| **Checkpoint** | **Software Design and Engineering** | **Algorithms and Data Structures** | **Databases** |
| Name of Artifact Used | React.js Card Game: War | React.js Card Game: War | React.js Card Game: War |
| Status of Initial Enhancement | Refactored components, improved game logic, and added animations. | Improved shuffle algorithm with Fisher-Yates, added tie-breakers. | Created database schema for storing player records. |
| Submission Status | In progress | Completed and submitted | Still working on it |
| Status of Final Enhancement | Planning final UI updates and animations. | Finished logic improvements, just testing now. | Still connecting backend for storing player stats. |
| Uploaded to ePortfolio | Not yet | Not yet | Not yet |
| Status of Finalized ePortfolio | Still working on it | Still working on i | Still working on it |