

| Reference Story Number | Reference Pool | Story Source             | Story Summary  |
|------------------------|----------------|--------------------------|--|
| 1                      | 1 Points       | EECS 581: Project 2      | <u>Minesweeper Easy AI</u> : Simulated an simple AI assistance using Python for Minesweeper that randomly uncovered cells and takes turns with the user to complete the game.  |
| 2                      | 1 Points       | EECS 581: Project 1      | <u>Minesweeper Setup</u> : Initialized Minesweeper using Python and PyGames data structures and components to lay a foundation for game implementation, by using pygames to create the basic grid UI layout.   |
| 3                      | 1 Points       | EECS 581: Project 1      | <u>GitHub Setup</u> : Created a comprehensive GitHub README page highlighting project purpose, features, installation steps, usage examples, and contribution guidelines to enhance clarity, engagement, and collaboration.  |
| 4                      | 1 Points       | EECS 581: Project 3      | Presentation: Created and designed a slide deck showcasing the core features of Elysian, clearly communicating its functionality, user experience, and future plans. Also, created a short demo video of our app.  |
| 5                      | 1 Points       | SELF Enterprise          | <u>Business Proposal Pitch</u> : Created rigorous documentation that defined the scope, timeline, budget, and motivation for the software project. This acted as an anchor for the rest of the phases in creating the end-product.                                     |
| 6                      | 2 Points       | EECS 678: Quash Project  | <u>I/O Redirection Part 1</u> : Implemented input redirection in C using the < operator, enabling programs to read data directly from files instead of standard input, improving automation and flexibility.   |
| 7                      | 2 Points       | EECS 678: Quash Project  | <u>I/O Redirection Part 2</u> : Implemented output redirection in C using the > operator, allowing program output to be written directly to a file, overwriting existing content for streamlined data management.  |
| 8                      | 2 Points       | EECS 678: Quash Project  | <u>I/O Redirection Part 3</u> : Implemented output redirection in C using the >> operator, enabling programs to append output to existing files without overwriting content, enhancing logging, persistence, and data tracking.  |
| 9                      | 2 Points       | Hackathon 2024: Wishmate | <u>Embedded AI into UI Framework</u> : Integrated the Amazon AWS Rekognition API to take pictures of items with the Kivy UI application to upload the pictures and store them in MongoDB.  |
| 10                     | 2 Points       | EECS 581: Project 1      | <u>Code Optimization</u> : Refactored code to ensure appropriate and optimized logic which helped improve code efficiency and runtime, such as deleting unnecessary variables and creating reusable functions.   |
| 11                     | 2 Points       | EECS 348: Final Project  | <u>Documentation</u> : Created the project plan, software requirements, software architecture, test cases, user manual, and project glossary documents for the arithmetic calculator to ensure all phases were properly recorded and used as reference in later steps. |

| Reference Story Number | Reference Pool | Story Source                         | Story Summary  |
|------------------------|----------------|--------------------------------------|--|
| 12                     | 2 Points       | EECS 581: Project 3                  | <u>Seamless User Experience:</u> Designed an intuitive and engaging UI/UX that prioritizes smooth user interactions while maintaining efficiency. Integrated multiple tools and thoughtfully experimented with color palettes to create a cohesive/user-friendly experience.   |
| 13                     | 3 Points       | Hackathon 2024: Wishmate Project     | <u>MongoDB Database:</u> Created a database with multiple collections to store user information and allow efficient querying of data. Integrated to web app using Python scripts.  |
| 14                     | 3 Points       | EECS 348: Final Project              | <u>Quality Assurance Testing:</u> Developed over 60 comprehensive test cases to validate all possible algebraic expressions, including edge cases, syntax errors, and complex user inputs, ensuring accuracy and reliability.  |
| 15                     | 3 Points       | EECS 348: Final Project              | <u>Parsing the Input for the Calculator:</u> Programmed functions to handle various different cases of user input for the calculator, including both bad and good input, by parsing the string input and tokenizing each into a data structure to pass into other files in the project.  |
| 16                     | 3 Points       | SELF Community Service Project (CSP) | <u>Subscription Management Database:</u> Utilized PostgreSQL to automate user data updates, enabling the generation of timely notifications and ensuring real-time communication, accuracy, and efficiency across the system.  |
| 17                     | 3 Points       | EECS 581: Project 3                  | <u>User Profile Data Storage:</u> Implement endpoints to store user profile data. This requires that a User Profile table be created in the database and an associated script to add data to the table.  |
| 18                     | 5 Points       | EECS 447: Semester Project           | <u>Database Architecture:</u> Designed database architecture using ER diagrams and logical models to define relations and entities that helped model a library system. It allowed for different tiers of members, creating accounts, renting and returning items, monitoring status of books, and charging accounts for overdue fees.          |
| 19                     | 5 Points       | EECS 581: Project 2                  | <u>Minesweeper Hard AI:</u> Simulated a sophisticated AI assistance using Python for Minesweeper than strategically uncovered cells, mimicking human intelligence. It flags mine cells, uncovered safe cells, and identified 1-2-1 patterns to make moves.   |
| 20                     | 5 Points       | EECS 447: Semester Project           | <u>Creating Queries:</u> Designed a set of SQL queries for the Library Database Management System to handle core operations such as inserting new books, updating availability after checkouts, searching records by title or author, joining user and book tables to track issued items, etc. while ensuring accurate data retrieval process! |
| 21                     | 5 Points       | SELF Community Service Project (CSP) | <u>Subscription Management React App:</u> Developed a full-stack subscription management module using React as the frontend framework, enabling dynamic user interactions, efficient state management, and seamless integration with backend services.   |

| Reference Story Number | Reference Pool | Story Source                            | Story Summary   |
|------------------------|----------------|---|---|
| 22                     | 5 Points       | EECS 581: Project 3                     | <u>Build the Model Architecture:</u> Create the parameters and imports needed for TensorFlow, initializing the model, and setting up the basics for it to start training on the Kaggle dataset.   |
| 23                     | 8 Points       | Personal Project: Heartland Linguistics | <u>Cutomer Centered UI:</u> Developed a website for a small business based on customer feedback and requirements. Highlighted the main critical parts of the website by using animations and graphics. Added navgation bar features and home page to help aquire customer attention, using TailwindCSS.                       |
| 24                     | 8 Points       | EECS 581: Project 2                     | <u>Minesweeper Medium AI:</u> Simulated a medium AI assistance using Python for Minesweeper that had some intelligence. It flags mine cells by calculating the number of covered adjacent cells and comparing it with the cell's number. It then uncover cells remaining cells that were not flagged, indicating a safe cell. |
| 25                     | 8 Points       | Hackathon 2024: TraWell                 | <u>Website UI:</u> Integrated new styles for the frontend of the website to help enhance user experience. Utilizes tailwind CSS to incorporate features such as responsive layouts, smooth hover transitions, modern card components, adaptive color schemes, and consistent spacing.   |
| 26                     | 8 Points       | EECS 678: Quash Project                 | <u>Run Executables:</u> Enabled the execution of programs with command-line parameters, searching the directories in the PATH variable if no absolute or relative path is given.  |
| 27                     | 8 Points       | EECS 581: Project 3                     | <u>Render Backend:</u> Integrated Render's backend into our GitHub app to create an interactive app where the user inputs get processed by our model. Created a tflite compressed model for the backend and endpoints in our app to call the backend.   |
| 28                     | 13 Points      | Hackathon 2024: Wishmate                | <u>AWS AI Implementation:</u> Utilized AWS Image Rekognition API along side Open CV to create an interaction feature for Wishmate. It allows users to take pictures and analyze the objects in the picture. Webscraping was used with this feature to look for associated Amazon links for the objects in the picture.        |
| 29                     | 13 Points      | San Francisco Hackathon                 | <u>AI Land Recommender:</u> Built an AI-driven recommendation engine that analyzes user preferences and farmland data to suggest optimal plots for fractional ownership on the Flow blockchain.   |
| 30                     | 13 Points      | Hackathon 2025: DreamDragon Project     | <u>Interactive UI:</u> Created a interactive UI using HTML, CSS, and JavaScript for all parts of the web app, including the game component. This required designed graphics from scratch, specific for each page. Later, it was integrated into our Flask app.  |

| Reference Story Number | Reference Pool | Story Source            | Story Summary  |
|------------------------|----------------|-------------------------|--|
| 31                     | 13 Points      | EECS 678: Quash Project | <u>Foreground and Background Processes:</u> Quash supports both foreground and background execution, where & indicates a background job. When a background job starts or completes, Quash displays messages showing the job ID, process ID, and command.                       |
| 32                     | 13 Points      | EECS 581: Project 3     | <u>Initial Project Setup:</u> Initialized the Expo Go app with React, Tailwind CSS, and Firebase backend. Integrating the three components into a initial working app and making sure it was compatible with new MacBooks, iOS devices, and Firebase base's Authentication DB. |