Manish Acharya

manish.acharya313@gmail.com | 0482013532
acharyamanish006.github.io | linkedin.com/in/manish-acharya3

About Me

Passionate software engineer with expertise in JavaScript, TypeScript, Node.js, and a wide range of web technologies. I am fascinated with compilers and interpreters and enjoy building them to deepen my understanding of programming languages.

"What I cannot create, I do not understand." – Richard Feynman

Skills

Languages: C/C++, Java, Python, JavaScript, TypeScript, Node.js, Go
Front-End: React.js, Next.js, HTML5, CSS3, Sass, Tailwind CSS
Back-End: Node.js, Express.js, Go, Flask, Go-Fiber
Containerization: Docker
Version Control: Git, GitHub
Database: MongoDB, MySQL
Other: RESTful APIs, Webpack, Babel, npm/yarn, Jest, Enzyme, JIRA, Agile/Scrum, AWS, WordPress.

Education

- **Bachelor of Information Technology (BIT) Sydney:**
  - **King's Own Institute (2023-2026):**
    - Relevant Coursework:
      - Object Oriented Programming, Web Design and Development.
      - Database Design and Development.
      - Systems Analysis and Design, Discrete Maths.

Experience

- **Member of KOISS (King's Own Institute Student Society):**
  - Collaborated with fellow members to organize and execute events such as workshops, seminars, and social gatherings, enriching the student experience.
  - Actively involved in community outreach programs and volunteer efforts organized by the society, contributing to the betterment of both campus and local communities.
  - Assisted in creating visually appealing event posters and promotional materials for the Student Society, utilizing graphic design skills to effectively communicate event details and attract attendees.

Project Work

- **MERN Ecommerce Application:**
  A full-stack Ecommerce application built with the MERN (MongoDB, Express.js, React.js, Node.js) stack. The project replicates major functionalities found in popular e-commerce websites, including user authentication, product management, shopping cart, and more. GitHub_Link

  - **Key Features:**
    - **User Authentication & Security:**
      - Account creation and secure login with email and password.
      - Passwords are securely hashed and stored.
      - Secure authentication with JSON Web Tokens (JWT).
      - Measures implemented to prevent common security vulnerabilities.

    - **Product Management & User Interaction:**
      - Product search functionality.
      - Users can create product listings with images, descriptions, and prices.
      - Detailed product view for users.
      - Shopping cart and wishlist functionality.
      - Users can add reviews and ratings for products.

    - **Database:**
      - MongoDB is used for storing user data, product listings, and order history.
      - Secure and efficient database interactions.

  - **Technologies Used:**
    - **Frontend:** React.js, CSS, Redux for state management.
    - **Backend:** Node.js, Express.js, MongoDB.
    - **Authentication:** JSON Web Tokens (JWT).

  - **Other Tools and Libraries:** Axios, bcrypt for password hashing, multer for file uploads.

- **LiteScript Programming Language and Interpreter:**
  LiteScript is an ongoing project dedicated to developing a programming language and its interpreter using TypeScript. This language is crafted to be lightweight and user-friendly, offering a simplified syntax while retaining powerful capabilities. GitHub_Link

  - **Key Features:**
    - **Lexer Development:**
      - Designed and implemented a robust lexer capable of tokenizing LiteScript code, and categorizing identifiers, keywords, numbers, and symbols.
      - Created the Token class and Token_Types enum to encapsulate token information.
      - Implemented a sophisticated parsing mechanism to decipher the structure of LiteScript code accurately.

    - **Parser Implementation:**
      - Developed a parser responsible for interpreting LiteScript tokens, identifying language constructs such as variable declarations and function definitions.
      - Implemented dedicated methods for parsing specific language features.

    - **Interpreter Script:**
      - Crafted an interpreter script showcasing the lexer and parser in action, demonstrating the recognition of LiteScript constructs.
      - Conducted thorough testing to ensure accurate interpretation and processing of

LiteScript code.
- Implemented dedicated methods for parsing specific language features.

• Simp (Simple Mathematical Interpreter):

Simp is a simple mathematical interpreter written in JavaScript. It supports basic arithmetic operations including addition, subtraction, multiplication, and division. The lexer parses the input expression into tokens, and the parser evaluates the expression respecting the correct order of operations. **GitHub_Link**

- **Key Features:**
  - **Lexer Development:**
    - Implemented a lexer that parses the input expression into tokens, categorizing numbers, and operators.
    - Developed token classes to encapsulate token information and ensure accurate tokenization of input.
  - **Parser Implementation:**
    - Created a parser that interprets the tokens produced by the lexer, respecting the correct order of operations.
    - Implemented methods for parsing and evaluating basic arithmetic operations (addition, subtraction, multiplication, division).
  - **Expression Evaluation:**
    - Designed the interpreter to handle arithmetic expressions and compute the result accurately.
    - Ensured that the interpreter respects operator precedence to produce correct results.

• ByteMix (JavaScript Game Library):

ByteMix is a lightweight and versatile JavaScript game library designed to simplify game development with a focus on performance and flexibility.**GitHub_Link**

- **Key Features:**
  - **Game Loop:**
    - Implements a robust game loop that initializes, updates, and renders the game state efficiently.
  - **Sprite Management:**
    - Provides utilities for loading and managing sprite images.
    - Facilitates easy drawing and manipulation of sprites within the game.
  - **Collision Detection:**
    - Includes collision detection mechanisms for both static and dynamic game objects.
    - Supports window boundary collision detection.
  - **Event Handling:**
    - Simplifies event handling, such as responding to user inputs.
  - **Gravity and Movement:**
    - Adds gravity to game objects for realistic movement.
    - Allows easy implementation of object movement and interaction.

Awards and Certificates

- **AWSome Day Online Conference(2023):**
  - Successfully completed the AWSome Day Online Conference, gaining in-depth knowledge of Amazon Web Services (AWS) cloud computing solutions.
  - Acquired insights into best practices for cloud architecture, security, and scalability.
  - Applied acquired skills in practical scenarios through hands-on sessions and case studies.