**Sameer Swain**

Hinjilicut, Odisha, India sameerswain.gse.23@outlook.com +917608076411  linkedin.com/in/sameer-swain-28b17b23b  [https://www.sameerswain.me](https://www.sameerswain.me/)

**SUMMARY**

Highly skilled software development engineer with expertise in both frontend and backend technologies. On the frontend, proficient in HTML5, CSS3, JavaScript (ES6+), Sass, Bootstrap, Webpack, Redux, React, Typescript, Angular, Git, Browser Developer Tools, Web Performance Optimization, Cross-Browser Compatibility, UX/UI Design Principles, RESTful APIs, Testing, Debugging, and CanvaPro.

On the backend, experienced in working with Node.js, Express.js, Socket.io, MongoDB, Mongoose, PostgreSQL, Redis, Firebase, AWS Lambda, S3, DynamoDB, Kubernetes, Docker, Data Structures, Algorithms (DSA), Design Patterns (DP), Low-Level Design (LLD), High-Level Design (HLD), and more.

Passionate about delivering high-quality, scalable software solutions, maintaining code integrity, and optimizing performance. Demonstrated ability to work autonomously or collaboratively within cross-functional teams. Strong problem-solving, communication, and leadership skills enable successful project execution.

Constantly keeping up with the latest industry trends and technologies, always eager to learn and apply new knowledge. Committed to delivering exceptional user experiences and contributing to the success of complex projects.

**PROJECTS**

**Frontend Development**

Bittobyte • <https://github.com/sdeSameerSweety/sdeSameerSweety.github.io> • August 2023 - August 2023

• In my awesome project called SameerSweetyFolio, which is my portfolio project, I utilized HTML, CSS, Bootstrap, and JavaScript to bring it to life. Let me break down what I accomplished in each aspect:  
• HTML:  
• HTML was the foundation of my project. With HTML, I created the structure and layout of the entire portfolio. Each page, including Home, About me, Skills, Experiences, Projects, and Contact me, was built using HTML. It allowed me to organize the content and define the different sections of my portfolio.  
• CSS:  
• To make my portfolio visually appealing and stylish, CSS was instrumental. I used CSS to apply various styles, including colors, fonts, spacing, and positioning. It helped me achieve the desired look and feel, making my portfolio visually engaging and professional.  
• Bootstrap:  
• Bootstrap came to the rescue when it came to creating a responsive and mobile-friendly design. By leveraging Bootstrap's framework and components, I ensured that my portfolio looked fantastic on different devices and screen sizes. It saved me time and effort, as I could easily integrate pre-built, responsive elements into my project.  
• JavaScript:  
• JavaScript added interactivity and dynamic functionality to my portfolio. I used JavaScript to implement features like animations, sliders, and navigation menus. It brought life to my portfolio and enhanced the overall user experience. It was great to showcase my skills and demonstrate my proficiency in this powerful scripting language.

**Backend Developemnt**

Functionup Bootacamp • <https://github.com/isamir909/Project-5-Ecommerce-Site> • September 2022 - September 2022

• FEATURE I - User APIs:  
- POST /login: Woohoo! I successfully implemented the login functionality, allowing users to authenticate with their email and password. When users provide their credentials, I verify them and send back their userId and a cool JWT token for successful authentication.  
- PUT /user/:userId/profile: I'm proud to say that I worked on the profile update feature. Users can modify their profile information by specifying their userId in the URL and sending their desired changes in the request body. Only the rightful user can update their own profile thanks to the authentication and authorization requirements I implemented.  
• FEATURE II - Product APIs:  
- GET /products: This one was really fun! I created the functionality to retrieve products. When users make a GET request to this API, they get a list of awesome products. I even added some nifty query parameters, so users can filter products by size, name, or price range. It's an excellent way for users to browse and find the products they love!  
• FEATURE III - Cart API:  
- PUT /users/:userId/cart: Oh, I'm proud of this one! I built the functionality to manipulate the user's cart. Users can remove products or decrease the quantity of a specific product. To make it work, they just need to provide their userId in the URL and specify the changes in the request body. It's fantastic to help users manage their shopping cart the way they want.  
• FEATURE IV - Order API:  
- POST /users/:userId/orders: Oh boy, I've been busy! I also worked on this feature, but I would love to know more details about the specific functionalities and requirements you had in mind. Usually, an order API involves creating an order based on the user's cart, doing some cool calculations, and updating essential data like inventory and order history. Let me know more about this feature—I'm here to help!  
• Oh, and testing! How could I forget? I proudly took part in testing too. It's crucial to ensure that everything works smoothly. I made sure the various API endpoints and their functionalities behaved as expected. Testing is the key to delivering a top-notch product!

**Backend Developemnt**

Functionup Bootacamp • <https://github.com/isamir909/Project-4-Url-Shortner> • August 2022 - September 2022

• Worked on two APIs: POST API for creating short URLs and GET API for redirecting to original URLs.  
• Implemented the logic for generating a short URL in the POST API.  
• Validated and formatted the URL code, and validated the long URL provided in the request body.  
• Returned the shortened unique URL as the response for the POST API.  
• Implemented redirection to the original URL in the GET API.  
• Used appropriate HTTP status codes for both successful redirection and error scenarios.  
• Actively participated in testing the APIs using Postman.  
• Created a Postman collection named "Project 4 Url Shortener" with separate requests for each API.  
• Ensured tests were functional and coordinated with team members for their tests.  
• Explored caching mechanisms to minimize database calls in Phase II.  
• Conducted research on using caching during redirection from shortened URLs for further optimization.

**Backend Developemnt**

Functionup Bootacamp • <https://github.com/isamir909/Project-3-BookManagement> • August 2022 - August 2022

• 1. Worked on two APIs: PUT and DELETE  
• - Implemented the PUT API to update book details such as the title, excerpt, release date, and ISBN.  
• - Handled unique constraints to ensure data integrity.  
• - Checked the existence of the bookId in the collection and whether the book is not deleted. Returned appropriate responses.  
• - Implemented the DELETE API to mark a book as deleted.  
• - Verified the existence and deletion status of the bookId. Returned suitable responses accordingly.  
• 2. Authentication and Authorization  
• - Implemented authentication for all book routes to ensure they are protected.  
• - Added a JWT token containing the userId, exp, and iat upon successful login.  
• - Validated the token before allowing access to protected routes, returning appropriate error messages for failed validations.  
• - Ensured proper authorization by allowing only the owner of a book to create, edit, or delete it.  
• - Implemented checks for unauthorized access and provided suitable error messages.  
• 3. Testing  
• - Created a new collection in Postman named "Project 3 Books Management" for testing the APIs.  
• - Added individual requests in the collection, properly named for clarity.  
• - Ensured that all team members had their tests in a running state within the collection.

**Backend Developemnt**

Functionup Bootacamp • <https://github.com/isamir909/Project-2-Open-to-Intern> • July 2022 - August 2022

• During the Project 2 Internship, I actively contributed to the team's success by:  
  
• Developed a POST API to create college documents, including essential details such as name, full name, logo link, and isDeleted status.  
• Implemented a POST API to create intern documents, capturing information such as name, mobile number, email, and associated collegeId. Ensured a return of HTTP status 201 and provided the created document as the response.  
• Collaborated effectively with team members to maintain a single git branch, facilitating code synchronization and integration with the front-end application.   
• Conducted comprehensive testing of the APIs using a dedicated Postman collection called "Project 2 Internship."   
• Organized the collection with appropriately named requests for each API, adhering to the prescribed naming conventions (e.g., "Create College" and "Get College Details").   
• Ensured that all tests within the collection were in a running state, validating the functionality and performance of the implemented APIs.   
• By taking on these responsibilities, I significantly contributed to the project's development, successful integration with the front-end, and the validation of our work through thorough testing.

**Backend Developemnt**

Functionup Bootacamp • <https://github.com/isamir909/Project-1-Bloging-Site> • June 2022 - July 2022

• Developed the backend for a Blogging-Site project as a backend developer  
• Successfully implemented two APIs:  
• 1. Created author documents by sending POST requests to the `BASE\_URL/authors` endpoint. Created at least 5 authors.  
• 2. Created blog documents by sending POST requests to `/blogs` endpoint. Ensured the authorId in the request body was a valid authorId by checking against the authors collection. Created at least 5 blogs for each author.  
• Handled error cases effectively:  
• - Returned HTTP status 400 for invalid requests with a response body.  
• - Implemented PUT requests to `/blogs/:blogId` for updating blogs.  
• - Checked the existence of blogId and isDeleted status before making updates. Returned appropriate HTTP status codes and response bodies.  
• Implemented authentication and authorization features:  
• - Added a POST request to `/login` for author login using email and password. On successful login, returned a JWT token containing the authorId.  
• - Validated the JWT token before calling protected endpoints. Returned suitable error messages and HTTP status codes for failed validations.  
• - Set the validated token in the request header as `x-api-key` and used middleware for authentication.  
• Ensured proper authorization:  
• - Only allowed the owner of blogs to edit or delete them.  
• - Returned appropriate error messages for unauthorized access.  
• Conducted comprehensive testing during development:  
• - Created a Postman collection named "Project 1 Blogging".  
• - Included individual requests for each API, properly named for clarity.  
• - Ensured all team members' tests were in a running state within the collection

**EDUCATION**

**DIPLOMA IN CIVIL ENGINEERING**

Minor in COMPUTER PROGRAMMING • GANDHI SCHOOL OF ENGINEERING • BHABANDHA, BERHAPMUR, GANJAM, ODISHA, INDIA • 2019 • 6.4

**COURSEWORK**

**AWS SOLUTION ARCHICTET**

NARESHIT INTSTUTE • Computer architecture and organization, operating systems, and networking • 2023

• Understanding computer architecture and organization helps in comprehending the underlying infrastructure of AWS, such as the virtualization techniques used, hardware components, and distributed systems. Knowledge of operating systems is valuable for grasping the concepts of system-level management and resource allocation in AWS, especially when configuring instances, managing storage, and optimizing performance. Networking expertise plays a significant role in comprehending AWS networking services, such as VPC (Virtual Private Cloud), subnets, routing, load balancing, and security groups, enabling efficient communication between resources. Applying computer architecture and organization principles assists in designing scalable and resilient AWS solutions, including selecting the appropriate EC2 instance types, managing database configurations, and optimizing resource utilization. Understanding operating systems and networking concepts allows for the implementation of robust security measures, enabling secure access control, data encryption, and network isolation within AWS Solution Architecture.

**SKILLS**

**Frontend: HTML5, CSS3, JavaScript (ES6+), Sass, bootstrap, Webpack, Redus, react, Typescript, Angular, Git, Browser Developer Tools, Web Performance Optimization, Cross-Browser Compatibility, UX/UI Design Principles, Knowledge of RESTful APIs, Testing and Debugging, CanvaPro.**

**Backend: Node.js, Express.js, Socket.io, MongoDB, Mongoose, PostgreSQL, Redis, Firebase, AWS Lambda, S3, DynamoDB, Kubernetes, Docker, DSA, DP, LLD, HLD.**