Manish Kumar Das

Full Stack Software Developer | Web Graphics

Github: https://github.com/the-halfbloodprince
Codepen: https://codepen.io/the-halfbloodprince
CV: CV - SDE.pdf

What's this?
A CV is limited and needs to be well presentable rather than well-detailed. A personal website is well-detailed and I'm working on one, but with the plethora of interesting and cool websites out there, I'm taking some time working on one that I won't start disliking after 2 weeks. So I'm presenting my work as a (not very detailed, but more than enough to get the gist) doc which I believe represents my work a lot better.

About Me

I'm an undergraduate student from the department of geology and geophysics at IIT Kharagpur, enrolled in its Integrated Master's degree in Applied Geology. I'm a boarder of the Radhakrishnan Hall of Residence and I'm from Jorhat, Assam.

I'm currently actively pursuing software development for the majority of my time, primarily Web Development. The spirit of treating imagination and creativity as challenges and finding ways of solving them has been one of the fundamental attributes of mine. It's in a way, my underlying algorithm to keep thinking about what can be done and how can it be done and that's what keeps me going.

Apart from that I am really fascinated and obsessed by the way software solves problems in such effective ways and has been able to substantially increase the quality of life over the last two decades. But most importantly, I find it quite a bit of fun and I don't see any boundaries to which this can expand, the possibilities seem endless and I'm very obsessed with the idea of pushing it even further with all my might and discover amazing stuff on the way.

Frontend Work

Here are some of my work which I believe would be relevant for you:
Also, here's my Codepen for additional reference: the-halfbloodprince - CodePen

Mozaic

- https://mozaic-akatsuki.vercel.app
- ***Repo Link:*** https://github.com/the-halfbloodprince/Mozaic

- Video Demo: https://drive.google.com/file/d/1I7QmWG5TTfIY0JplG95zyHf5Py8zkTcq/view?usp=share_link
- This is an application my team built in a recent **hackathon**, in which we won the **first prize**.
- The application is **NOT RESPONSIVE** at the moment, as we implemented it in a limited timeframe for the hackathon and we had to focus more on the functionalities.
- I had fully contributed to creating the frontend of the application, especially all the functional bits.
- I worked on implementing ~95% of the frontend of the application and around ~10% of the landing page(Home).
- You may notice **significant slow loading speeds whenever any operation involving the blockchain is performed**, such as logging in, creating NFT, listing/unlisting NFT, buying/selling NFT. The app uses some free and easy platforms for blockchain and web app deployment which makes it quite slow at the moment, so **please be patient for the content to load even if it takes quite some time**.
- Some points to highlight about the app:
    - The initial pick for styling was TailwindCSS but later defaulted to **CSS Modules**.
    - We used **Mantine** as a component library to aid in fast development of complex components like **Drag and Drop** inputs in the **Create** page.
    - Used **React Contexts** for state management.

Xhopie

- Xhoppie (xhopie.vercel.app)
- ***Repo Link:*** https://github.com/the-halfbloodprince/xhopie
- This is an application I attempted as I was learning NextJS, Tailwind and Redux.
- This was implemented taking the help of various articles, blogs, videos and clones.
- The design was not pre-planned as it was initially just an attempt to learn the stack, so I decided to go with a design I found in a similar tutorial video, highly inspired by Amazon.
- This app features many useful functionalities such as:
    - The products data is NOT stored with us, it is fetched from the Fake Store API
    - Authentication with NextAuth
    - State Management with Redux Toolkit
    - Atomic CSS styling with TailwindCSS
    - A rough and simple payment system with Stripe, which is doesn't accept real card credentials at the moment, it needs test card credentials:
        - Card No: 4242 4242 4242 4242
        - Expiry Date: 04 / 24
        - CVC: 424
    - Backend routes with NextJS's api routes which support NodeJS.

Github Profile Dash

- https://github-profile-dash-cra.vercel.app/

- ***Repo Link:*** https://github.com/the-halfbloodprince/github-profile-dash-cra
- This is an application I built as a task for an internship which I could successfully grab.
- This involved user entering their GitHub username, which is then used to fetch their GitHub statistics using GitHub's API.
- This project involved the use of several key functionalities like:
  - **User Input Debouncing**: Binding the API call on each type by the user would cause a significant number of useless requests, hence the user's input was debounced to wait for a certain time interval before making the API call.
  - **Data Fetching with proper loading, error and data states handled**: Using axios and custom react hooks, I was able to abstract the data fetching into an interface exposing the required attributes like loading, error states etc.
  - **Usage of Reusable Component Library with Material UI**: This was one of the requirements of the task to use Material UI. Material UI exposes multiple reusable and customizable pre-built components to be used across the app maintaining consistent styling and single-stop configuration.
  - **Dark and Light theme**: Implementation of separate dark and light mode UIs using Material UI and custom states and properties.

Other Work

Stocklift

- ***Repo Link:*** https://github.com/the-halfbloodprince/kodeinkgp_hackathon_client

Stocklift.vercel.app

- https://stocklift.vercel.app/

3D Experiments

Perspectives

- ***Repo Link:*** https://github.com/the-halfbloodprince/Perspectives
- https://perspectives-psi.vercel.app/

Galaxy M1199

- ***Repo Link:*** https://github.com/the-halfbloodprince/GalaxyM1199
- https://galaxy-m1199.web.app/

Galaxy M2999

- ***Repo Link:*** https://github.com/the-halfbloodprince/Galaxy-M2999
- https://galaxy-m2999.web.app/

Gringotts MineCart Ride

- https://gringotts-mine-cart-ride.web.app/

Crystal-01

- ***Repo Link:*** https://github.com/the-halfbloodprince/Crystal-01
- Crystal (crystals-01.netlify.app)

Backend Work

Here are **some** of my past work (in **Backend**):

Vidrent

Repo Link: https://github.com/the-halfbloodprince/VidRent
This is a sample **video rental application** I built as an **hands-on learning project to enhance my knowledge and understanding of NodeJS** and using it to build a backend for web applications. Here are some key features implemented in the application:

1. Proper file structure for easy **readability** and navigation
2. Use of a NoSQL database **(MongoDB)** though this was a good application to use a relational database, but as mentioned I developed this project as a learning project so I chose the tech which I wanted to learn at that point.
3. Authentication with **JSON Web Tokens** and **Auth Middlewares** relying on a **bearer token**.
4. File based logging as well as logging of warnings and errors into MongoDB with **Winston**.
5. Proper **Data Validation** at every CRUD endpoint.

The tech stack used for this project:

1. NodeJS
2. ExpressJS
3. MongoDB
4. Mongoose (MongoDB ORM)
5. Winston (for logging)
6. JSON Web Tokens

This was project I **built as a complete beginner** to the Backend Development portfolio and hence inevitably **made a lot of mistakes** but still this is a project where I tried out and **implemented a wide arsenal of tools** and hence value it a lot as a showcase of my experiences with different tools and a revision template for a lot of key concepts.

But any project is mostly valued only when it is maintained to match up to new good practices and tools as time goes on. Unfortunately I haven't been able to devote much time to it but I'm aware of my future ambitions and here are my future goals regarding the same:

1. Integrate **TypeScript** for better type safety and features.
2. Shift most of the database work to a **SQL database for better performance and relevance**. **Data which doesn't require a SQL database (eg: error logs) will still remain in the NoSQL database**.
3. Improve **data validation with Zod and TypeScrip**t.
4. Improve Authentication with modern techniques, namely **OAuth**, and **Time Based OTP**, and others as we come along.
5. Develop a **Frontend** for the same.

Devprofile

Repo Link: https://github.com/dscnsec/devprofile/tree/master
This is the application I built during my participation in **Winter of Code** by DSC NSEC. This app has both client and server folders which are orchestrated together with concurrently, an npm package. I contributed to both the client as well as the server side and integrated it with **Github API** and **MongoDB**.

Xhopie

Repo Link: the-halfbloodprince/xhopie (github.com)
Xhopie is an e-commerce application I built again as a learning project to understand **NextJS**. NextJS provides **both a server and a client** to work on, and the backend logic required for the application has also hence been written in NextJS.
This application also uses **Stripe to mock test payments** but relies on the **FakeStore API** to fetch the products. It also implements saving successful test orders into **Firestore**.

Stocklift Server (*majorly co authored*)

Repo Link: the-halfbloodprince/kodeinkgp_hackathon_server (github.com)
This was an application I built out while being a part of a team participating in a hackathon. This is the repository specifically for the it's backend. Do note that the code is not well formatted or structured as it was built in strict time constraints for the hackathon and also I had only some basic guiding and helping out in the same, it was mainly built by my teammates when I was busy working on building a performant and good frontend.

Blogs / Articles

Stripes in Shaders

Link: https://dev.to/thehalfbloodprince/stripes-in-shaders-3n5n

Shaders in web graphics, particularly in frameworks like WebGL and Three.js, are specialized programs that control the rendering pipeline to enhance visual effects. These scripts manipulate the appearance of 3D objects by defining how light interacts with surfaces, enabling dynamic and realistic graphics. Shaders play a crucial role in creating immersive and visually stunning web-based experiences.

This comprehensive article delves into shader programming, focusing on creating stripes in WebGL using GLSL which is a simple concept but encompasses a wide range of basic tricks and methodologies for WebGL beginners. It covers fundamental concepts such as normalized coordinates, simple gradients, and color combinations. The article demonstrates the manipulation of stripes through sine functions, frequency adjustments, and directional variations, providing valuable insights for aspiring shader developers.