

You can measure the speed of the subsea flow using the tilt angle of the rod by applying basic physics principles. The method involves analyzing the forces acting on the rod:

Method Overview:

1. Forces Acting on the Rod:

- Buoyancy Force (B)
- Gravitational Force (W)
- Drag Force (D) from the Flow

2. Equilibrium Condition:

- The rod will tilt until the torque due to the drag force is balanced by the torque from the weight and buoyancy.
- The drag force D is given by: $D = \frac{1}{2} C_d \rho V^2 A$ where:
 - C_d is the drag coefficient of the rod,
 - ρ is the density of seawater,
 - V is the flow velocity,
 - A is the projected area of the rod.

3. Deriving Flow Velocity V :

- By measuring the tilt angle θ , you can solve for V using equilibrium equations.
- Typically, the drag force and the gravitational force balance at an angle, and using trigonometry, you can derive V .

4. Implementation:

- Use an IMU (Inertial Measurement Unit) or an accelerometer to measure the tilt angle.
- Calibrate the setup in a controlled environment before deploying it in subsea conditions.
- Compare readings against known flow rates to derive a relationship between tilt and velocity.

Articles and Research Papers:

Let me find some relevant articles for you.

Measuring subsea flow velocity using the tilt of a rod is a recognized method known as the "drag-tilt principle." This approach involves deploying a device that tilts in response to water currents, with the tilt angle correlating to the flow speed.

Tilt-Current Meters (TCMs):

Tilt-Current Meters are instruments designed to measure flow speed and direction by utilizing the drag-tilt principle. A notable example is the TCM developed by Lowell Instruments, LLC. This device measures flow speed and direction by recording the tilt caused by water currents. The recorded tilt data can be converted into flow speed and direction.

hurricaneisland.net

Research and Applications:

A study titled "Measuring water current speed and direction of a long term underwater sensor (LoTUS) using tilt and roll compensation" by Marcus Lindkvist explores a method to compute water current speed and direction using an accelerometer and magnetometer. The results indicate that this method can be used for scientific data collection.

diva-portal.org

Additionally, tilt current meters have been utilized in various environments, including coastal waters and the surf zone, to measure stationary current velocities. Their low cost and simplicity make them suitable for such applications.

coastaldynamics2017.dk

Implementation Considerations:

- **Sensor Selection:** Choose a tilt sensor suitable for subsea conditions. MEMS-based inclinometers are often preferred due to their full 360° range and rapid response time. For instance, the SUBSEA-2 series inclinometers are designed for underwater applications and can measure angles in both pitch and roll axes while submerged.
leveldevelopments.com
- **Calibration:** Calibrate the system in a controlled environment to establish a relationship between tilt angle and known flow velocities.
- **Data Logging:** Ensure the system can record tilt angles over time, allowing for analysis of flow variations.
- **Environmental Factors:** Consider factors such as buoyancy, tether length, and deployment depth, as they can influence the tilt response.

By carefully selecting appropriate sensors and calibrating the system, you can effectively measure subsea flow velocities using the tilt of a rod or similar device.

Tell me about yourself

Sure! I'm a Full Stack Web Developer with over ten years of experience building web applications and working on everything from front-end interfaces to back-end systems. Over the past few years, I've been focusing a lot on blockchain technology, creating smart contracts, and deploying cloud-based solutions on platforms like AWS.

I've worked on some interesting projects, like developing decentralized apps and optimizing workflows for better performance. I enjoy problem-solving and finding practical ways to use technology to make things more efficient or accessible.

Outside of work, I like to stay updated on new tech trends and experiment with different tools. and I'm really excited about the opportunity to contribute to something innovative like Spexi's platform.

Tell me about your latest role

In my current role as a Software Engineer at CMOrq, a Web3 DeFi company, I've been working on developing blockchain-based web applications. A big part of my work involves writing and deploying smart contracts and building decentralized systems.

I've also managed deployments on AWS, using tools like Terraform for infrastructure as code, which helps ensure scalability and reliability. One of my favorite parts of the job has been streamlining workflows through automation—like setting up CI/CD pipelines with GitHub Actions, which has sped up our deployment process significantly.

The projects I work on often focus on making decentralized technologies more accessible, and I really enjoy the mix of problem-solving and innovation it brings. It's been a great experience learning and contributing to cutting-edge Web3 technology.

Why are you leaving your current role?

I've really enjoyed my time at CMOrq, especially working on innovative Web3 projects and seeing the impact of our work in the DeFi space. Most of the major projects I've been involved in are now completed, and the team is currently in a bit of a quieter phase.

While it's been great to have a moment to reflect, I'm eager to take on new challenges and contribute to exciting, high-impact projects. That's why I'm really drawn to Lazer – the chance to collaborate with a talented team and build the next generation of decentralized applications is exactly what I'm looking for.

Do you review about us?

Yes, I reviewed your company, XLOOP. I'm impressed with how you focus on leveraging AI to enhance operational efficiency and data-driven decision-making. Your solutions seem innovative, especially in optimizing complex workflows and driving automation in dynamic industries.

The blend of cutting-edge technology with a clear focus on practical business outcomes resonates with my experience in integrating technology to solve real-world problems. I'd love to learn more about your recent initiatives and how I can contribute as a Tech Leader to further your mission.

Why are you interested at us?

I'm particularly interested in XLOOP because of your innovative approach to leveraging AI for optimizing workflows and delivering tangible business value. Your focus aligns closely with my passion for using technology to drive efficiency and create scalable solutions.

As someone with a strong background in full-stack development, cloud infrastructure, and blockchain, I see tremendous potential in applying my expertise to your mission. The opportunity to lead a team and contribute to developing advanced solutions excites me, especially at a company like XLOOP that's at the forefront of combining AI with cutting-edge technologies.

Beyond that, your emphasis on innovation and collaboration reflects a culture where I can thrive and make a meaningful impact.

React experience

I've been working with React for several years, and it's one of my go-to tools for building dynamic and interactive user interfaces. I love how its component-based architecture allows me to create reusable and modular pieces of code, which improves both development speed and maintainability.

For state management, I've worked with Redux, Context API, and more recently, React Query, depending on the complexity of the application. I also focus heavily on performance optimization, using techniques like lazy loading, memoization, and code-splitting to ensure fast and seamless user experiences.

When it comes to integration, I've connected React front-ends to both REST and GraphQL APIs, ensuring smooth data interactions. Testing is another area I prioritize—using tools like Jest and React Testing Library for unit testing, and Cypress for end-to-end scenarios.

Overall, I enjoy how flexible React is and how its ecosystem lets me deliver scalable, high-quality solutions tailored to user needs. I'm always looking for ways to leverage it to improve the user experience and meet business goals.

Cloud infrastructure experience

I have extensive experience designing and managing cloud infrastructure, primarily on AWS and Azure, to build scalable, secure, and efficient systems. In my previous roles, I've worked on setting up cloud environments from scratch, automating deployments, and optimizing existing infrastructure for performance and cost.

On AWS, I've used services like EC2, S3, RDS, and Lambda extensively, along with tools like CloudFormation and Terraform to implement infrastructure as code. I've also integrated CI/CD pipelines using tools like AWS CodePipeline, Jenkins, and GitHub Actions to streamline deployments and ensure reliable delivery.

With Azure, I've managed virtual machines, containers using AKS, and serverless workflows. I've also worked on migrating on-premises applications to the cloud, ensuring a smooth transition with minimal downtime.

Beyond that, I've focused heavily on security and monitoring. I've implemented IAM policies, network segmentation, and cloud monitoring tools like AWS CloudWatch and Azure Monitor to ensure systems are secure, highly available, and optimized for performance.

Overall, I enjoy working with cloud platforms because they provide the flexibility and scalability to meet the needs of modern applications, and I'm always looking for ways to enhance infrastructure efficiency and reliability.

Node experience

I've been working with Node.js for several years, building scalable back-end systems and APIs for a wide range of applications. I appreciate its non-blocking, event-driven architecture, which makes it ideal for handling high-concurrency scenarios and real-time applications.

I've used Node.js to develop RESTful and GraphQL APIs, integrating them seamlessly with front-end applications and databases like MongoDB, PostgreSQL, and MySQL. I'm proficient in using popular frameworks like Express.js and NestJS to streamline development and maintain a clean, modular codebase.

For asynchronous operations, I've worked with promises, `async/await`, and event-driven patterns to ensure smooth performance. Additionally, I've implemented authentication and authorization systems using JWT, OAuth, and session management for secure user interactions.

I'm also experienced in building microservices using Node.js, leveraging Docker and Kubernetes to deploy and manage these services at scale. I've integrated message brokers like RabbitMQ and Kafka for event-driven architectures.

Testing and quality are key for me, so I frequently use tools like Mocha, Chai, and Jest for unit and integration tests. I also prioritize performance optimization, using tools like PM2 and profiling to ensure efficient execution in production.

Node.js is an essential part of my toolkit for creating fast, reliable, and scalable back-end systems that meet both technical and business requirements.

Smart contract experience with EVM-compatible, gas-efficient contracts for Proof of Capture, Data License-to-Access NFTs, and tokenization.

In my recent projects, I've had hands-on experience developing EVM-compatible smart contracts, particularly focused on creating gas-efficient contracts for various use cases like Proof of Capture and tokenization.

For example, I've worked on smart contracts that manage NFTs, specifically for license-to-access scenarios, where the NFT serves as a key to unlock certain data or functionality. I've been mindful of gas optimization in these contracts, especially when dealing with on-chain data or transactions that could otherwise become costly.

In addition, I've built systems for Proof of Capture, ensuring the data's authenticity and linking it securely to blockchain transactions. This process involves creating gas-efficient methods for verifying and recording data while keeping transaction costs low.

I also spent time optimizing contract functions and ensuring that operations like minting, transferring, and validating tokens or NFTs happen smoothly without consuming too much gas. This involved strategies like using batch processing or minimizing state changes within the contract.

Leverage IPFS, Filecoin, or similar decentralized storage systems to manage geospatial data effectively.

I've worked with decentralized storage systems like IPFS to manage and store large amounts of data securely and efficiently. Specifically, for geospatial data, using IPFS allows us to store high-resolution imagery and related files in a way that's decentralized and resilient. I've integrated IPFS into blockchain-based applications to ensure the data's availability while keeping it tamper-proof and transparent.

For example, in a previous project, I helped set up a storage system where aerial images were uploaded to IPFS, and the metadata or access information was stored on-chain. This setup made it easy for users to retrieve data, verify its authenticity, and track its history. We also explored Filecoin as a way to add additional layers of redundancy and incentive for data storage, which can be crucial for long-term accessibility of large datasets.

Leveraging these technologies ensures that geospatial data is not only stored in a secure and distributed way, but also that it's easily accessible for end-users or applications that need it.

Solidity and smart contract experience

I've worked extensively with Solidity for writing smart contracts, particularly in the context of building decentralized applications. A lot of my experience has been focused on developing gas-efficient contracts, including those for tokenization, NFT minting, and data validation.

For example, I've written contracts for managing NFTs, ensuring efficient minting, transfers, and the integration of metadata in a way that minimizes transaction costs. I've also developed contracts for various Web3 applications, such as Proof of Capture systems, where data like images or videos are verified and stored in a way that guarantees authenticity.

A key part of my approach is optimizing for gas efficiency—whether it's through using the right data types, reducing on-chain computations, or batching transactions. I've also spent time reviewing and auditing contracts to ensure they're secure and function as expected.

I stay up-to-date with the latest Solidity updates and best practices to make sure the contracts I write are both secure and scalable.

Web3 Infrastructure and privatekey management experience

I've worked extensively with Web3 infrastructure, primarily focusing on deploying and managing decentralized applications (dApps) and integrating blockchain networks. My experience includes setting up and interacting with Ethereum-based systems, working with smart contracts on EVM-compatible chains, and managing cross-chain interactions. I've also dealt with layer-2 scaling solutions to improve transaction speed and reduce gas fees.

In terms of private key management, I've implemented secure solutions for handling private keys both in development and production environments. This includes using secure wallet solutions like MetaMask and hardware wallets, along with integrating key management practices into our Web3 applications. I've also utilized tools like Infura and Alchemy for interacting with the Ethereum network while ensuring key security, especially when automating transactions or managing user accounts.

I'm very mindful of keeping private keys secure by following best practices—such as using encrypted storage, avoiding storing keys on the client side, and leveraging multi-signature wallets or hardware wallets for high-value operations. It's a crucial part of ensuring both the security and integrity of any Web3 system.

Tech challenges

I've encountered several tech challenges in backend development. One notable challenge was at CMORQ, where I was tasked with deploying a new blockchain application using AWS. We faced issues with scalability due to the high volume of transactions, which led to performance bottlenecks.

To tackle this, I optimized our architecture by implementing a microservices approach and leveraging AWS services like Lambda for serverless computing and DynamoDB for efficient data storage. This significantly improved our app's responsiveness and allowed it to handle increased loads without crashing.

=====

Happy New Year! I hope you had a great start to the year. I'm really excited to be here today and looking forward to our conversation.

Tell me about yourself

I'm a tech-savvy Full Stack Web Developer with a strong passion for building scalable and innovative solutions. Over the past decade, I've gained hands-on experience working with front-end and server-side technologies, as well as Web3 ecosystems.

At CMOrq, I developed blockchain web applications, integrated smart contracts, and managed AWS deployments with Terraform, all while streamlining CI/CD pipelines to improve efficiency. Before that, at Innovative Solutions, I worked on cloud-focused projects like transcription automation pipelines using AWS and data migrations from Azure to AWS.

What really excites me about this role at Lazer is the opportunity to create decentralized applications and stay at the forefront of Web3 tech. I'm also confident my experience with frameworks like React, Ethers.js, and AWS, combined with my knack for solving complex problems, makes me a great fit.

Outside of work, I'm always tinkering with emerging technologies, and I love exploring how blockchain can solve real-world challenges.

Tell me about your latest role

In my current role at CMOrq, I've been focusing on developing blockchain-based web applications for a leading Web3 DeFi company. My work involves integrating smart contracts, deploying scalable solutions on AWS using Terraform, and implementing GitHub CI/CD pipelines to streamline our deployment process.

One project I'm particularly proud of was optimizing cloud-based workflows and improving performance throughout the development lifecycle. It not only enhanced our team's efficiency but also ensured the applications remained reliable and scalable for end users.

What excites me most about this role is the constant opportunity to innovate within the Web3 space and build decentralized solutions that solve real-world problems.

Why are you leaving your current role?

I've really enjoyed my time at CMOrq, especially working on innovative Web3 projects and seeing the impact of our work in the DeFi space. Most of the major projects I've been involved in are now completed, and the team is currently in a bit of a quieter phase.

While it's been great to have a moment to reflect, I'm eager to take on new challenges and contribute to exciting, high-impact projects. That's why I'm really drawn to Lazer – the chance to collaborate with a talented team and build the next generation of decentralized applications is exactly what I'm looking for.

Did you review us?

Yes, absolutely! I've taken some time to learn about Lazer, and I'm really impressed by what the company stands for. I love how you've built a world-class team with talent from top companies like Apple, Google, and Coinbase, and how you've been able to help startups grow their products from \$0 to \$200M revenue.

What excites me most is your focus on building innovative digital products and the emphasis on deeply understanding client needs. It's clear that Lazer is not just about delivering solutions but also about creating lasting value through expertise in emerging tech like Web3.

What are you looking for in your next role?

I'm looking for a role where I can continue building innovative Web3 solutions while pushing the boundaries of what decentralized technologies can do. I want to work on impactful projects that challenge me technically and creatively, and I value being part of a team that's passionate about delivering exceptional results.

I'm also looking for an environment where I can grow alongside talented peers, share ideas, and contribute my skills to drive success. A role that allows me to stay at the forefront of Web3 trends and collaborate with clients to solve real-world problems is exactly what I'm after.

Experience with React, Node, Next.js

I have extensive experience working with React, Node.js, and Next.js in various roles. For example, at Ignite-Reading, I refactored key components from Ruby on Rails to React, improving performance and maintainability. I also developed reusable components and a shared library to streamline front-end development across projects.

With Node.js, I've built APIs and backend services, ensuring they're scalable and efficient. At Innovative Solutions, I used Node.js alongside AWS Lambda to create serverless solutions that handled large-scale data operations.

Next.js has been a go-to framework for me when building modern, server-side-rendered applications. It's been incredibly useful for optimizing performance and SEO, especially for client-facing projects.

Experience with Web3 libraries and frameworks on EVM and Solana

I've worked extensively with Web3 libraries and frameworks on EVM chains and have some exposure to Solana as well. On the EVM side, I've used tools like Web3.js and Ethers.js for interacting with smart contracts, handling wallet integrations, and querying blockchain data. For example, at CMOrq, I developed decentralized applications where I integrated these libraries to interact seamlessly with existing smart contracts.

I'm also familiar with The Graph for querying blockchain data efficiently, and tools like Hardhat and Truffle for testing and deploying smart contracts. My experience extends to managing decentralized file storage with IPFS and implementing node interactions with frameworks like Alchemy and Moralis.

On Solana, I've explored its ecosystem, particularly in building dApps and understanding its unique architecture, including the use of Solana web3, metaplex, magic eden and etc.

Proven experience in web3 dApp development across the stack

I have solid, proven experience in Web3 dApp development across the full stack. For instance, at CMORq, I was responsible for designing and developing decentralized applications on EVM chains, where I handled both the front-end and smart contract integrations. I used React and Next.js for the front-end, while interacting with smart contracts using Web3.js and Ethers.js on the back-end. I also worked with AWS for deployment and scaling, ensuring the dApps were both robust and secure.

My experience extends to building APIs that interact with blockchain data and ensuring smooth communication between the Web3 and Web2 layers. Additionally, I've integrated storage solutions like IPFS for decentralized file management, and used The Graph for querying blockchain data.

I'm comfortable with both the smart contract side of things, using Solidity for contract development and deployment, as well as the full application development, from implementing the UI/UX designs to managing back-end services.

AWS and Azure experience

I have extensive experience working with both AWS and Azure, leveraging their cloud services to build scalable, efficient solutions.

On the AWS side, I've worked with a variety of services including EC2, Lambda, S3, RDS, DynamoDB, and API Gateway. For example, at Innovative Solutions, I designed and implemented a logging system using DynamoDB, significantly reducing access time by 60%. I also worked on serverless architecture using AWS Lambda, S3, and API Gateway to automate transcription pipelines. Additionally, I've used Terraform for infrastructure as code, ensuring that our AWS infrastructure is scalable and maintainable.

With Azure, I have experience with services like Azure Functions, Azure Storage, and Azure Active Directory, which I used during a data migration project from Microsoft Dynamics to AWS. I also implemented APIs using Azure's API Management service, allowing seamless integrations with other cloud services.

My experience with both platforms has given me a deep understanding of cloud infrastructure, scalability, and cost optimization across different environments.

What is your desired salary?

I would expect a salary in the range of \$8,000 to \$10,000, depending on the overall compensation package, including benefits and potential growth opportunities.

I'm open to discussing this further to ensure we're aligned and that we find a number that works for both of us.

Tech challenges

I've encountered several tech challenges in backend development. One notable challenge was at CMORQ, where I was tasked with deploying a new blockchain application using AWS. We faced issues with scalability due to the high volume of transactions, which led to performance bottlenecks.

To tackle this, I optimized our architecture by implementing a microservices approach and leveraging AWS services like Lambda for serverless computing and DynamoDB for efficient data storage. This significantly improved our app's responsiveness and allowed it to handle increased loads without crashing.

=====

Tell me about yourself

I'm a web and mobile developer with over 10 years of experience in building scalable and high-performance applications. I've worked with a range of technologies, like React, Next.js, Angular, and Node.js for the frontend and backend. I also have hands-on experience in mobile app development for both iOS and Android using tools like React Native, Xamarin, and Flutter.

In my past roles, I've led teams to deliver impactful projects, whether that's optimizing web apps for performance, reducing deployment times with CI/CD pipelines, or improving user engagement through thoughtful design. I've worked across different industries, including e-commerce and high-traffic web apps, and I thrive in fast-paced, agile environments.

I'm all about delivering high-quality code and making sure the user experience is top-notch. On the side, I like to mentor teams, share knowledge, and solve problems in creative ways. So yeah, that's me in a nutshell!

Tell me about your latest role

In my latest role at MCG(Merchants' credit guide company), I led cross-functional teams to develop high-performance web and mobile applications for clients across various industries. It was a pretty exciting job because I got to work with both React and Next.js for web apps, which was all about creating fast, dynamic user experiences. I also worked on mobile apps using

Xamarin and React Native, helping reduce time-to-market by 30% while keeping the performance and user experience close to native.

One of my favorite parts was spearheading the implementation of CI/CD pipelines, which cut deployment times by 50%, allowing us to push out updates faster and more frequently. I also focused a lot on performance optimization, like fine-tuning database queries and indexing to improve data retrieval speeds.

Collaboration was a huge part of my day-to-day, especially with design and product teams. We were able to improve UI/UX, which boosted customer satisfaction and increased user retention by 25%. Overall, it was a role where I got to wear multiple hats—leading teams, optimizing performance, and ensuring we were building scalable, maintainable solutions for our clients.

Did you review about us?

Yes, I've looked into Barber-OS, and I'm really excited about what you're doing! Your mission to transform the barbershop and salon industry through innovative software and hardware is inspiring. It sounds like a unique opportunity to work on applications that directly impact the customer experience and streamline operations.

I also love that you emphasize a collaborative culture and inclusivity in the workplace—that's something I really value. The fact that you offer flexibility, professional growth, and even an on-site gym shows that you care about your team's well-being and development. I think the blend of working on cutting-edge tech and contributing to a mission with such a tangible impact is a perfect fit for me. I'm excited about the chance to join a team that's as passionate about technology and customer experience as I am!

Why join us?

I'd love to join Barber-OS because of the opportunity to work on projects that truly make a difference in an industry that's ripe for innovation. Your focus on transforming barbershops and salons really speaks to me, as it's a unique space where tech can have a huge impact on both customer experience and business efficiency.

I'm also drawn to your collaborative and inclusive culture. I thrive in environments where teamwork and creativity are encouraged, and I can tell that Barber-OS fosters that kind of atmosphere. The flexibility you offer in work hours and the focus on work-life balance is another big plus for me.

From a technical perspective, I'm excited by the chance to work with technologies like Django, React, and React Native, and I'm eager to contribute my skills to building robust, performance-driven applications. Plus, the opportunity for professional growth and being part of an innovative team is something I'm really looking for in my next role. It's all about making an impact while growing alongside an amazing team!

Why did you leave your last position which took 6 years?

I left my last position after six great years because I was seeking more flexibility in my work and an opportunity to dive into new technologies. After a solid run at MCG, I realized that I wanted a change to focus more on my personal growth, especially in the field of AI, which is one of the biggest trends in tech right now. The role at MCG gave me a lot of amazing experiences, but I felt it was time to explore new challenges, particularly in an environment that offered more flexibility and allowed me to further develop my skills in AI.

Taking this time off has allowed me to explore new interests, expand my knowledge, and get ready for a role where I can bring all of that into play. Now, I'm excited to find a place like Barber-OS, where I can blend my background in development with the chance to explore innovative technologies in a flexible, forward-thinking environment.

Why are you coming back to web development not, ai?

I'm still actively learning AI and diving deeper into that field, but I realized that the gap in my career could potentially set me back. While I'm passionate about AI, I didn't want to let this gap grow too wide, especially since I've built a strong foundation in web and mobile development over the past 10 years. I believe that jumping back into a web development role will help me maintain momentum in my career, especially in an environment like Barber-OS, where I can leverage my skills in full-stack development while also continuing to explore AI on the side.

It's all about balancing my long-term goals with the immediate opportunity to get back into a role that will keep me moving forward, professionally and technically. And who knows, the experience I gain in this role might even open up more ways to apply AI to web development in the future!

What are you looking for in your next role?

In my next role, I'm really looking for a chance to work on exciting, impactful projects that push me to grow both technically and creatively. I want to be in an environment that values collaboration and innovation, where I can contribute my expertise in web and mobile development, but also continue learning, especially in areas like AI. Flexibility in work hours and the opportunity to have a healthy work-life balance are also important to me. I'm looking for a team that's passionate about what they do, where I can collaborate closely with others and contribute to building something meaningful.

Additionally, I'd love to be part of a company like Barber-OS, where the mission is both innovative and practical, and I can see the direct impact of my work. I'm excited about the chance to continue growing as a developer while also helping the team build scalable, performance-driven solutions that make a difference in the real world.

Django experience

I've got solid experience with Django, having used it extensively in my previous roles. One of my key projects was at IBASE, where I developed a robust backend system using Django to support high-traffic web applications. I focused on optimizing query performance, implementing caching strategies, and integrating third-party services via Django REST Framework.

I've also used Django for building RESTful APIs, handling authentication and authorization, and ensuring secure user management with its built-in features. I'm comfortable working with Django ORM for managing database interactions and have experience fine-tuning performance to ensure scalability and efficiency.

Overall, I'm very familiar with Django's full-stack capabilities, from setting up the initial project structure to deploying applications. It's one of the frameworks I'm most comfortable with, and I'm always looking for ways to leverage its strengths to build powerful, efficient backends.

Python experience

I have extensive experience with Python, both in web development and automation. In my role at IBASE, I used Python extensively with Django to build backend systems, focusing on optimizing database interactions and implementing caching strategies. I also integrated Python with third-party services and APIs, helping to streamline workflows and improve performance.

Beyond Django, I've used Python for scripting, data processing, and automation tasks. For example, at AMI, I developed Python scripts to automate order processing and real-time inventory updates, which significantly improved operational efficiency. I've also worked with Python for backend services that support real-time updates, ensuring smooth user experiences.

Python is one of my go-to languages, and I enjoy using it to solve complex problems, whether it's for backend development, automation, or integrating with other technologies.

React experience

I have a lot of experience working with React, both in web and mobile development. Over the years, I've used React extensively to build dynamic, high-performance web applications. For example, at MCG, I worked on developing web applications with React and Next.js, focusing on improving user engagement and performance. I implemented features that reduced page load times by 40%, which helped create a smoother and more responsive user experience.

I'm also very comfortable working with state management tools like Redux, which I've used to manage complex app states in both web and mobile applications. At AMI, I led the conversion of multiple e-commerce sites from ASP.NET to React, which resulted in a 40% performance boost.

React's component-based architecture is something I really enjoy working with, as it allows for modular, reusable code. I'm always focused on making sure the apps are fast, responsive, and easy to maintain, while also ensuring a great user experience across different devices.

React Native experience

I have solid experience with React Native, particularly in building cross-platform mobile applications. At MCG, I worked on several mobile projects using React Native, helping to reduce development time and maintain a native-like performance across both iOS and Android platforms. One of the standout projects was developing a mobile app where we integrated real-time features and optimized for performance to ensure a smooth user experience.

Additionally, I've used React Native to build apps that integrate with native modules and third-party libraries. For example, I've worked with custom native modules for handling device-specific functionality like camera access and push notifications, which helped improve the app's overall user experience.

React Native's ability to share code between platforms while still offering a near-native experience is something I find really exciting. It's a great way to speed up development while still delivering a high-quality product, and I'm always looking for new ways to improve the performance and scalability of React Native apps.

Redux experience

I've worked extensively with Redux, especially for managing state in complex React and React Native applications. In my role at MCG, I used Redux to handle the state of dynamic web applications, making sure that the data flow was predictable and maintainable across large, feature-rich apps. I implemented middleware like Redux Thunk for handling asynchronous actions, ensuring smooth data fetching and state updates without performance bottlenecks.

At AMI, I led the effort to integrate Redux into multiple projects, helping the team manage complex states in e-commerce and inventory management systems. We used Redux to centralize and synchronize state across components, which made the app more scalable and easier to debug. I also worked closely with the team to improve the performance of the app by optimizing how state was updated and preventing unnecessary re-renders.

Overall, I've found Redux to be a great tool for maintaining clear and efficient state management, especially in large applications with a lot of user interactions and complex data flows. I'm comfortable with both basic implementations as well as more advanced patterns, and I enjoy helping teams optimize their state management for better performance and scalability.

Django REST framework experience

I have extensive experience working with Django REST Framework (DRF), particularly in building robust APIs and integrating them with front-end and mobile applications. In my role at IBASE, I developed several RESTful APIs using DRF to support high-traffic web applications. These APIs were optimized for performance, with careful attention to things like query optimization, pagination, and caching to ensure fast, reliable data retrieval.

I've also worked on securing APIs by implementing authentication and authorization strategies using DRF's built-in tools like token-based authentication, OAuth, and permissions. This was especially crucial for applications with sensitive data or complex user roles.

At AMI, I used DRF to integrate third-party services and external APIs, which extended the functionality of the applications I was working on. For instance, I integrated ERP and inventory management systems via DRF, allowing real-time updates and improving the user experience.

I'm very comfortable with DRF's serializers, viewsets, and routers, which allow me to quickly build APIs that are easy to maintain and scale. Overall, I've found Django REST Framework to be an excellent tool for building efficient and secure APIs that integrate seamlessly with both front-end and mobile applications.

REST APIs experience

I have significant experience working with REST APIs, both in developing and consuming them. In my previous roles, I've used REST APIs to integrate various services and enhance application functionality.

At IBASE, I developed RESTful APIs using Django REST Framework to support backend services. I focused on creating efficient, secure endpoints that could handle high traffic and return data quickly. I implemented things like pagination, filtering, and query optimization to ensure scalability and performance. I also handled API versioning and worked with clients to ensure the APIs met their requirements and were easy to consume.

Additionally, I've integrated third-party REST APIs into multiple projects. For example, at AMI, I integrated e-commerce systems and real-time inventory APIs, improving the overall functionality of the app. I used Python and JavaScript to handle asynchronous data fetching, ensuring smooth communication with external services.

I'm familiar with the full lifecycle of REST API development, from designing endpoints to handling authentication and error management. I also have experience with testing and documenting APIs, which helps ensure they are reliable and easy to work with for both developers and clients.

SQL experience

I have strong experience working with SQL, particularly in managing databases and optimizing queries for performance. Throughout my career, I've worked with both relational and NoSQL databases, but SQL has been a core part of my backend development.

At IBASE, I was responsible for managing and optimizing large relational databases. I worked with complex queries, ensuring they were efficient by using proper indexing, joins, and optimizing query execution plans. I also used SQL to design and manage schemas, ensuring data integrity and performance. For example, I implemented indexing strategies to reduce query times, which helped with the performance of high-traffic applications.

In my role at AMI, I also worked with SQL databases to support real-time inventory updates and automated order processing. I wrote optimized queries to fetch and update data quickly, which was critical to ensuring a smooth user experience in the e-commerce platform.

In addition to my hands-on experience, I'm comfortable working with SQL tools and techniques like joins, subqueries, stored procedures, and data migration. I've used these skills to design efficient data models and handle large-scale data operations with ease. I'm always focused on ensuring that the database layer is scalable, secure, and performant.

Version Control, containerization and Ci/CD experience

I have solid experience with version control, containerization, and CI/CD, all of which are essential for modern development workflows.

Version Control: I've been using Git for version control throughout my career, managing codebases for both individual and team projects. I'm comfortable with Git branching strategies like feature branches, Git flow, and trunk-based development. At MCG, I worked closely with the team to set up Git workflows that allowed us to collaborate efficiently on multiple parallel tasks and ensure high-quality code delivery. I've also used GitHub and GitLab for code reviews, pull requests, and issue tracking to streamline development and ensure best practices in version control.

Containerization: In terms of containerization, I've used Docker extensively to create lightweight, portable, and reproducible environments for both development and production. I've set up Docker containers for microservices, ensuring that each service runs in an isolated environment. This was especially useful when working on backend services at IBASE, where we needed to ensure consistency across different stages of development and deployment. I've also used Docker Compose to manage multi-container applications, making it easier to spin up environments for local development or testing.

CI/CD: For CI/CD, I've set up and maintained continuous integration and delivery pipelines using tools like Jenkins, GitLab CI, and CircleCI. At MCG, I was responsible for setting up the CI/CD pipeline, automating the build, test, and deployment process, which resulted in a 50%

reduction in deployment times and more frequent releases. I've worked on automating tests, linting, and deployment to staging and production environments, ensuring that the process is smooth and reliable.

Overall, I've found that version control, containerization, and CI/CD practices are crucial for maintaining high-quality, scalable code while also improving team productivity and reducing manual errors in the deployment process.

CSS, HTML, JS experience

I have extensive experience with CSS, HTML, and JavaScript, which are the foundational technologies I use for building responsive and interactive web applications.

CSS:

I've used CSS extensively to design and style web applications, focusing on making them visually appealing, consistent, and responsive. I'm proficient in using modern CSS techniques, including Flexbox, Grid, and custom properties (CSS variables). I also have experience with CSS preprocessors like SASS, which helps with organizing and maintaining styles in large applications. For example, at MCG, I used CSS to create responsive layouts and optimize the UI for different screen sizes, improving the user experience across desktop and mobile platforms. I'm also familiar with frameworks like Bootstrap and Tailwind CSS, which I've used for rapid prototyping and building scalable, maintainable designs.

HTML:

HTML is a core skill I use for structuring content on the web. I've worked with semantic HTML to ensure accessibility and SEO best practices, making sure that the applications I build are user-friendly and perform well in search engines. I've also implemented forms, tables, and multimedia elements, ensuring they work seamlessly across all browsers. At AMI, I focused on building interactive and user-friendly e-commerce sites, using HTML to structure the pages and integrate dynamic content.

JavaScript:

JavaScript is at the heart of the interactive features I build in both web and mobile applications. I'm proficient in vanilla JavaScript for tasks like DOM manipulation, event handling, and asynchronous programming (using promises and async/await). Additionally, I've worked with JavaScript libraries and frameworks like React, Next.js, and Angular, where I use JavaScript to power the dynamic, component-driven UIs. For example, at MCG, I implemented client-side features using JavaScript to create responsive, fast-loading pages that integrated with the backend APIs. I'm also experienced with working on front-end performance optimization, such as lazy loading and minimizing JavaScript payloads.

Overall, I enjoy using CSS, HTML, and JavaScript to create seamless and engaging user experiences, while ensuring that the web applications are responsive, performant, and accessible.

Responsive design and cross-browser compatibility experience

I have significant experience with both **responsive design** and **cross-browser compatibility**, ensuring that the applications I build provide seamless user experiences across all devices and browsers.

Responsive Design:

I've implemented responsive design principles in various projects, ensuring that the layout and UI components adjust smoothly to different screen sizes and resolutions. I'm proficient with CSS techniques like **Flexbox** and **CSS Grid** to create fluid layouts that adapt to both small and large screens. For example, at MCG, I worked on dynamic web applications where I used media queries to adjust the layout based on the user's device, ensuring that everything from navigation menus to image sizes was optimized for mobile and desktop devices. I also utilized responsive design frameworks like **Bootstrap** and **Tailwind CSS** to speed up the development process, while still ensuring custom design adjustments to meet project requirements.

Cross-Browser Compatibility:

Ensuring cross-browser compatibility has been a critical part of my development workflow. I'm experienced in testing applications across multiple browsers (like Chrome, Firefox, Safari, and Edge) and making adjustments to ensure consistent styling and behavior. I've used tools like **Autoprefixer** to automatically handle vendor prefixes and ensure that my CSS works across older browsers. Additionally, I'm familiar with **polyfills** to handle missing features in some browsers and **feature queries** to conditionally apply styles based on browser capabilities.

At **AMI**, we had a large e-commerce platform where users accessed the site on different devices and browsers. I worked on ensuring that our design looked consistent and functioned well across these environments. This included testing for things like CSS rendering, JavaScript performance, and ensuring interactive elements (forms, buttons, etc.) worked consistently across browsers.

Overall, I prioritize responsive design and cross-browser compatibility in every project, making sure that the user experience remains smooth, accessible, and consistent no matter how or where the application is accessed.

=====

Tech interview with Pragya

I want to start by sincerely apologizing for missing our last meeting and for the inconvenience it caused. I truly appreciate your patience and the opportunity to reschedule.

I'm looking forward to our conversation today and am fully prepared to make the most of our time together.

=====

Tell me about yourself

Hey! So, I'm Ibec, a Full Stack Web Developer with a pretty solid background in both front-end and back-end development, plus a lot of hands-on work in Web3 and blockchain tech. I've been diving into the blockchain world lately, working on decentralized finance applications and smart contracts, mainly using Solidity. In my current role at CMORQ, I'm building Web3 apps and handling deployments on AWS with Terraform, so scalability and automation have become second nature for me.

Before that, I worked with cloud services at Innovative Solutions, where I set up logging systems, automated transcription pipelines, and migrated data to AWS. And honestly, I love problem-solving, especially when it comes to optimizing performance or setting up smooth CI/CD pipelines that make life easier for everyone involved.

On a more personal note, I'm really into exploring new tech, especially anything around DeFi and blockchain, and I'm all about building things that have a big impact. I think that's what drew me to this role—your focus on 1-click checkout and the potential for really impactful, efficient tech is exciting. So yeah, that's a bit about me!

Tell me about your latest role

Sure! So, in my current role at CMORQ, I'm a Software Engineer focused on building decentralized applications in the Web3 DeFi space. A lot of what I do involves working with smart contracts, mainly in Solidity, to create blockchain-based solutions. I've also been leading the deployment efforts on AWS, using Terraform to manage infrastructure as code, which has been a game-changer for keeping things scalable and reliable.

One of my main focuses has been optimizing our CI/CD pipelines with GitHub Actions, which has really helped speed up our deployment process and cut down on any bottlenecks. I'm also always on the lookout for ways to streamline our workflows and boost performance at every stage of development—whether that's by tweaking the cloud setup or refining the deployment steps.

Overall, this role has been a mix of hands-on coding, infrastructure management, and pushing for smoother project workflows, which I've really enjoyed. It's kept me in the loop with the latest in Web3 tech and given me some great experience with scaling blockchain applications, which I think aligns well with what you're looking for!

Tell me about your Innovative Solutions experience

Yeah, my time at Innovative Solutions was pretty awesome. I was a Full Stack Cloud Engineer there, and most of what I worked on involved cloud infrastructure, automation, and improving efficiency with AWS. One of the biggest projects I handled was setting up a logging system using AWS DynamoDB, PHP, and Typescript, which ended up reducing access time by about 60%—that was a huge win for performance.

I also got to automate transcription processes with AWS services like S3, Lambda, and Transcribe, which cut down maintenance costs. Another fun project was designing a client website using Vue.js and Laravel, where I could dig into the front-end and back-end. Plus, I led a data migration from Microsoft Azure Dynamics to AWS RDS, so I worked a lot with APIs and AWS Lambda to handle data flows.

Overall, it was a lot of cloud architecture, automation, and getting hands-on with different tech stacks, which kept things really interesting!

Did you review about our company?

Absolutely, I took some time to look into your company and I'm really impressed! I love that you're focused on creating the most efficient 1-click checkout system; it's such a game-changing concept in the e-commerce space. The fact that you've already secured \$10M in funding and are generating some solid ARR shows you've got a strong foundation to build on.

I also noticed that your team is passionate about innovation and pushing the boundaries of what's possible with blockchain technology, which resonates with me a lot. I'm excited about the opportunity to contribute to such a forward-thinking project and work with a team that's really making an impact in the Web3 space. Plus, the remote work flexibility is a huge bonus for me! Overall, I think it's a fantastic fit, and I'm eager to learn more about what you're building.

Why join us?

Great question! I'd love to join your team because I see a lot of exciting potential in what you're doing. Your focus on building a 1-click checkout solution really stands out to me; it's such an innovative approach that can transform the online shopping experience.

I'm also drawn to the fact that you have a strong funding base and are already generating revenue. That kind of stability gives me confidence that I'd be part of a team that's not just dreaming big but also making it happen.

I'm passionate about Web3 and blockchain technology, and being part of a company that's at the forefront of this space feels like a perfect fit for my skills and interests. Plus, I love collaborating with others who are just as excited about innovation as I am. I think my experience in developing decentralized applications and optimizing processes aligns well with your goals,

and I'm eager to contribute to the exciting work you're doing. It just feels like a great opportunity to grow and make a real impact!

Why are you leaving your current role?

Good question! While I've really enjoyed my time at CMORQ and have learned a ton about building decentralized applications and working with blockchain technology, but most of the big projects there are pretty much wrapped up, so right now I'm mainly focused on bug fixes and maintenance. I'm looking for something that'll let me dive into more impactful, challenging work again. I'm particularly excited about the opportunity to focus on protocol engineering and tackle the complexities of things like liquidity pools and cross-chain smart contracts, which I don't get to dive into as much in my current role.

I believe that joining your team will allow me to leverage my skills in a new way and contribute to an innovative project that aligns with my passion for Web3. So, I feel like it's time for me to explore new horizons and continue my professional journey in a direction that excites me!

What are you looking for a new role?

I'm looking for a new role that challenges me and allows me to dive deeper into the blockchain and Web3 space. I'm really eager to work on complex projects like cross-chain smart contracts and liquidity pools, which are areas I find super fascinating.

I also want to be part of a collaborative team that values innovation and is pushing the boundaries of technology. I thrive in environments where I can learn from others and share my knowledge, so a role that encourages teamwork and creative problem-solving is a big plus for me.

Ultimately, I'm hoping to find a position where I can contribute to exciting projects while also growing my skills and taking on new responsibilities. Your company seems like a great fit because of the focus on innovative solutions and the opportunity to make a real impact in the industry!

What are your strength and weakness?

Strength: "I'd say one of my biggest strengths is my ability to quickly adapt to new technologies and frameworks. I love diving into new challenges, and I've got a knack for picking things up fast, whether it's a new programming language or a complex concept in blockchain. I'm also a strong problem-solver; I enjoy breaking down challenges and finding efficient solutions, which has helped me streamline workflows and improve project outcomes.

Another strength is my experience in both front-end and back-end development. This full-stack background allows me to see the bigger picture and collaborate effectively with different team members, whether they're focused on design or infrastructure.

Weaknesses: "As for weaknesses, I'm looking to improve my public speaking skills. While I'm comfortable communicating with my team, I want to get better at presenting ideas in larger settings. I've started seeking out opportunities to practice this more, like participating in team meetings or small group presentations.

React Experience

I've worked extensively with React in several roles. At Ignite-Reading, for instance, I refactored key components from Ruby on Rails to React, which helped improve both performance and maintainability. I also built a component library for the team, using Storybook for documentation, which streamlined our front-end development and collaboration.

I've worked with tools like Redux for state management and have experience setting up optimized workflows using GitHub actions for CI/CD. React is my go-to for creating dynamic, user-focused interfaces, and I enjoy working with its ecosystem to deliver efficient, scalable front-end solutions.

TypeScript Experience

I have substantial experience with TypeScript, which I've used to enhance the robustness and maintainability of my code. At Innovative Solutions, I implemented a logging system using TypeScript alongside AWS services, which helped improve performance and reduce access times significantly.

I really appreciate how TypeScript adds type safety to JavaScript, making it easier to catch errors during development rather than at runtime. It's particularly helpful in larger projects, like when I was developing APIs or complex applications where keeping track of data types can get tricky.

Additionally, I've integrated TypeScript into React projects to benefit from its features while building components, which has improved both the development experience and the overall code quality. I enjoy the added structure it provides and how it can lead to more reliable and maintainable codebases.

Redux experience

I have solid experience with Redux, especially in managing complex state in React applications. At Ignite-Reading, I utilized Redux to handle global state management, which streamlined data flow across components and improved the overall performance of our application. I implemented actions, reducers, and middleware to manage side effects effectively. In addition to Redux, I've worked with other state management tools like Context API for smaller projects where a full Redux setup might have been overkill. I've also explored Zustand and Recoil for more recent projects, as they offer simpler APIs for managing state in React. Each tool has its strengths, and I like to choose the right one based on the project's needs and

complexity. I'm always open to trying new libraries to find the best solutions for state management challenges.

Tech challenges

I've encountered several tech challenges in backend development. One notable challenge was at CMORQ, where I was tasked with deploying a new blockchain application using AWS. We faced issues with scalability due to the high volume of transactions, which led to performance bottlenecks.

To tackle this, I optimized our architecture by implementing a microservices approach and leveraging AWS services like Lambda for serverless computing and DynamoDB for efficient data storage. This significantly improved our app's responsiveness and allowed it to handle increased loads without crashing.

Collaborate with frontend and backend developers to integrate smart contract functionalities experience

Working with front-end and back-end developers to integrate smart contract functionalities has been one of my favorite parts of the job. When we start a project, I usually sit down with the front-end team to understand their needs and how they envision the user interactions. We talk about what kind of data the smart contract will handle and what functions we need to expose.

On the back-end side, I coordinate with developers to make sure the APIs are set up correctly to communicate with the smart contracts. This usually involves defining the endpoints that will trigger contract functions and return data. I love how everyone brings their expertise to the table, and it's all about finding the best way to connect everything seamlessly.

During integration, I'm often debugging issues together with the teams. If something doesn't work as expected, we'll all hop on a call, go through the code, and troubleshoot the problems together. I find that collaboration really speeds things up and leads to better solutions since we're all aligned on the project goals.

Once everything is integrated, I help with testing to ensure that the front end is interacting correctly with the smart contract. It's exciting to see the whole system come together, especially when we can visualize the smart contract actions in the UI. That collaborative energy makes the process enjoyable and ensures we deliver a smooth user experience!

Perform code reviews, testing, and debugging of smart contracts

Sure! In my experience with code reviews, testing, and debugging smart contracts, I focus on ensuring quality and security. During code reviews, I dive into the logic of the contract, checking for vulnerabilities and efficiency. For example, I once spotted an unnecessary function call that could've increased gas fees, so I suggested a more streamlined approach that saved costs.

When it comes to testing, I write unit tests to cover various scenarios and edge cases, using frameworks that integrate well with Rust and Solana. If issues arise, I dig into transaction logs and recreate the problem locally to troubleshoot effectively. For instance, I encountered a bug in a lending contract that caused unexpected results during transactions, and by collaborating with my teammates, we identified and fixed the logic error, ensuring everything ran smoothly on the mainnet.

Node experience

I have solid experience with Node.js, which I've used extensively for building server-side applications. At CMORQ, I developed blockchain web applications and managed the deployment on AWS using Node.js for the back end. I've worked with various frameworks, such as Express, to set up APIs and handle server logic efficiently.

In my previous roles, I also implemented serverless architectures using AWS Lambda, which allowed me to create scalable applications without worrying about server management. I'm comfortable integrating Node.js with databases, handling real-time data with WebSocket, and implementing middleware for routing and authentication. Overall, Node.js has been a crucial part of my tech stack, and I enjoy leveraging its capabilities to build robust, high-performance applications.

PHP Experience

I've been working with PHP for years, mainly in environments where I need to build scalable and maintainable APIs and handle high-traffic loads. I'm comfortable with frameworks like Laravel and Symfony, which I've used to create RESTful APIs, handle authentication, and work with data processing.

I focus on writing clean, reusable code and I prioritize speed and reliability. PHP's evolved a lot with each version, so I've adopted modern practices—like dependency injection and namespaces—along with newer features in PHP 8 that improve performance. MySQL integration is a big part of my PHP projects too, and I've used ORM libraries and raw SQL based on performance needs.

Most importantly, I keep performance and security in mind throughout, especially when scaling.

MySQL Experience

With MySQL, I've been involved in everything from schema design to optimizing complex queries. I've designed normalized databases for scalability and performance but am also comfortable denormalizing when it makes sense for speed. Indexing is something I pay close attention to; I always check to ensure my indexes align with query patterns, especially for high-traffic applications.

For query optimization, I use `EXPLAIN` to analyze and tune SQL statements and avoid pitfalls like full table scans. I also implement caching layers where necessary, either in MySQL or through tools like Redis, depending on the needs of the project.

Handling data integrity and security is crucial too, so I regularly work with transactions and ensure proper isolation levels. Plus, I'm familiar with setting up replication for load balancing and backups, and I've worked with sharding in cases where MySQL has to handle really large datasets.

=====

Router infrastructure experience

To be honest, I don't have direct experience with router infrastructure specifically listed on my resume. However, I'm very enthusiastic about learning and I'm confident in my ability to pick up new skills quickly.

I have experience working with AWS services and managing API integrations, which has given me a solid understanding of how data flows through systems and how to optimize for performance. While I haven't worked directly with routing protocols, I'm eager to dive into that area and leverage my existing skills to understand the complexities involved.

I believe my problem-solving mindset and ability to adapt to new challenges will allow me to contribute effectively in this area. I'm really excited about the opportunity to learn and grow, especially in a role that focuses on innovative solutions in the blockchain space.

ERC-437, paymaster experience

I have hands-on experience with ERC-4337 and paymasters in my recent projects. I've worked on implementing the Account Abstraction feature that ERC-4337 introduces, which allows for more flexible transaction processing. This has been particularly beneficial in enhancing user experiences in decentralized applications by enabling functionalities like gasless transactions.

In my current role, I've been involved in developing smart contracts that utilize paymasters to manage transaction fees, which not only improves user onboarding but also helps to reduce friction for end-users interacting with our applications. This has given me a solid understanding of how paymasters can be integrated into broader systems to support various use cases in the DeFi space.

I'm really passionate about the potential of ERC-4337 and its ability to streamline user interactions within blockchain ecosystems, and I'm excited about the prospect of bringing this expertise to your team!

What projects did you build using ERC-4337 and paymasters?

I've worked on a couple of projects that leverage ERC-4337 and paymasters. One notable project was a decentralized finance application where we implemented account abstraction to simplify the user experience.

In this project, we used paymasters to allow users to perform transactions without needing to hold native tokens for gas fees. This meant users could interact with the app seamlessly, which significantly improved user onboarding and retention rates. The paymaster contract handled the transaction fees on behalf of the users, which we configured to support various payment methods, including third-party tokens.

Another project involved integrating ERC-4337 functionality into a wallet solution. We designed it to facilitate gasless transactions and enable features like social recovery and session management, all aimed at making it easier for non-technical users to interact with decentralized applications.

Overall, these experiences have given me a strong grasp of how ERC-4337 and paymasters can enhance user experiences in the Web3 ecosystem, and I'm excited about the potential to build even more innovative solutions moving forward!

LP experience

I have experience working with liquidity pools, particularly in the context of decentralized finance applications.

One of the key projects I worked on was a decentralized exchange (DEX) where I helped set up and manage liquidity pools. This involved creating smart contracts that governed the pooling of assets, ensuring that liquidity providers could earn rewards while also maintaining a balanced pool for users to trade against.

I also focused on optimizing the liquidity distribution and developing strategies to incentivize users to contribute to the pools, such as yield farming rewards. This experience has given me a solid understanding of the mechanics behind liquidity pools, including impermanent loss, and how they play a crucial role in enhancing the overall efficiency of DeFi protocols.

I'm really passionate about the dynamics of liquidity and how they can be leveraged to create robust decentralized ecosystems, and I'm excited about the opportunity to apply this knowledge to your projects!

Cross-chain smart contract experience

I have experience working with cross-chain smart contracts, particularly in projects that aim to enhance interoperability between different blockchain networks. In my recent role, I was

involved in developing a solution that allowed for seamless asset transfers and interactions across multiple blockchains.

One project I worked on focused on creating a bridge that facilitated token swaps between Ethereum and Binance Smart Chain. This required implementing smart contracts that could lock tokens on one chain while minting equivalent tokens on the other, ensuring security and reliability throughout the process.

I also explored using protocols like Polkadot and Cosmos, which emphasize cross-chain functionality. This involved writing and testing smart contracts that could communicate with different blockchain environments, enabling features like decentralized exchanges that operate across multiple chains.

Through these experiences, I've gained a solid understanding of the challenges and solutions involved in building cross-chain smart contracts, including security considerations and the importance of robust testing. I'm excited about the potential of cross-chain technology and look forward to applying this knowledge to future projects!

=====

Tell me about yourself

Sure! I'm a full-stack engineer with a mix of front-end, back-end, and cloud experience, plus a strong interest in web3 tech. I've been in the field for a while, working across different industries, from DeFi at CMORQ to more traditional web applications earlier on. Right now, I'm wrapping up at CMORQ, where I helped build blockchain applications and automated AWS deployments with Terraform.

I'm really into finding ways to make tech efficient and scalable—whether it's improving CI/CD pipelines to speed things up or optimizing cloud infrastructure. Outside of work, I'm always exploring the latest tech trends, which keeps me curious and ready to bring new ideas to my projects. Now, I'm looking for a new challenge where I can build impactful products, ideally with a team that values innovation and user-centered design.

Tell me about your latest role

Yeah, so in my current role at CMORQ, I'm working as a Software Engineer for a web3 DeFi company. It's been all about developing decentralized applications using blockchain tech. I've

been building smart contracts, managing cloud deployments on AWS, and automating CI/CD pipelines with GitHub. One of the coolest things I've worked on is optimizing blockchain-based applications for performance and scalability, especially when handling tons of transactions.

I also handle the infrastructure side, using Terraform for infrastructure-as-code to keep everything scalable and reliable. And since we're all about decentralized finance, security is a big focus—I'm always making sure everything's running smoothly and securely. It's been a great mix of tech challenges and problem-solving!

Why join us?

Well, first off, the fact that you've got a profitable platform with such a small, skilled team really says a lot—it's rare to see a company at your stage with that kind of efficiency. And I like that the company's all about supporting creators and brands in one ecosystem; it feels like there's so much room to innovate. Plus, I'm looking to jump into a place where I can work across both B2B and B2C products, so it seems like a solid fit for what I bring and what I want to build next.

Why are you leaving your current company?

Yeah, I'm still with CMORQ, but most of the big projects there are pretty much wrapped up, so right now I'm mainly focused on bug fixes and maintenance. I'm looking for something that'll let me dive into more impactful, challenging work again. ShopMy's growth and focus on the creator economy is really exciting to me, and I think it'd be a great place to bring my skills and really contribute to something meaningful.

What are you looking for in your next role?

I'm looking for a role where I can really get hands-on with new projects and make an impact right from the start. I'd like to be part of a team where there's room to innovate and tackle interesting challenges—especially in areas that connect tech and user experience, like your products. Working somewhere that values growth, both for the company and the people on the team, is really important to me too. I'm hoping to find a place where I can bring my experience but keep learning and growing alongside a solid team.

What are your strengths and weaknesses?

Strengths

One of my biggest strengths is my adaptability. I've worked with a variety of tech stacks and projects, so I can quickly pick up new languages and frameworks as needed. I'm also really good at problem-solving—whether it's optimizing performance or troubleshooting issues, I enjoy digging in and finding solutions. Plus, I'm a solid team player; I love collaborating with others, sharing ideas, and helping my teammates whenever I can.

Weaknesses:

As for weaknesses, I sometimes find myself getting a bit too caught up in the details when I'm working on a project. I aim for perfection, which can slow me down. I'm working on balancing that by setting clear priorities and deadlines to keep things moving. Another area I'm focusing on is improving my public speaking skills; I tend to be a bit shy in larger groups. I've been making an effort to participate more in team meetings to build my confidence.

Node experience

I have solid experience with Node.js, which I've used extensively for building server-side applications. At CMORQ, I developed blockchain web applications and managed the deployment on AWS using Node.js for the back end. I've worked with various frameworks, such as Express, to set up APIs and handle server logic efficiently.

In my previous roles, I also implemented serverless architectures using AWS Lambda, which allowed me to create scalable applications without worrying about server management. I'm comfortable integrating Node.js with databases, handling real-time data with WebSocket, and implementing middleware for routing and authentication. Overall, Node.js has been a crucial part of my tech stack, and I enjoy leveraging its capabilities to build robust, high-performance applications.

PHP Experience

I've been working with PHP for years, mainly in environments where I need to build scalable and maintainable APIs and handle high-traffic loads. I'm comfortable with frameworks like Laravel and Symfony, which I've used to create RESTful APIs, handle authentication, and work with data processing.

I focus on writing clean, reusable code and I prioritize speed and reliability. PHP's evolved a lot with each version, so I've adopted modern practices—like dependency injection and namespaces—along with newer features in PHP 8 that improve performance. MySQL integration is a big part of my PHP projects too, and I've used ORM libraries and raw SQL based on performance needs.

Most importantly, I keep performance and security in mind throughout, especially when scaling.

MySQL Experience

With MySQL, I've been involved in everything from schema design to optimizing complex queries. I've designed normalized databases for scalability and performance but am also comfortable denormalizing when it makes sense for speed. Indexing is something I pay close attention to; I always check to ensure my indexes align with query patterns, especially for high-traffic applications.

For query optimization, I use `EXPLAIN` to analyze and tune SQL statements and avoid pitfalls like full table scans. I also implement caching layers where necessary, either in MySQL or through tools like Redis, depending on the needs of the project.

Handling data integrity and security is crucial too, so I regularly work with transactions and ensure proper isolation levels. Plus, I'm familiar with setting up replication for load balancing and backups, and I've worked with sharding in cases where MySQL has to handle really large datasets.

=====

Did you review our company?

Yep, I've looked into about your company and all the unique ways you're helping creators and brands connect. Super interesting! That said, I'd really like to get a deeper understanding from your perspective—I'm sure you could give me more insight into what makes it tick!

I'm also impressed by your commitment to creating a frictionless experience for users in the financial sector. The emphasis on security and the use of smart contracts aligns well with the current trends in DeFi, and it's exciting to think about contributing to a platform that's changing the way credit is accessed globally. Plus, being part of a team that's at the forefront of AI and financial tech sounds like a fantastic opportunity to make a real impact!

React Experience

I've worked extensively with React in several roles. At Ignite-Reading, for instance, I refactored key components from Ruby on Rails to React, which helped improve both performance and maintainability. I also built a component library for the team, using Storybook for documentation, which streamlined our front-end development and collaboration.

I've worked with tools like Redux for state management and have experience setting up optimized workflows using GitHub actions for CI/CD. React is my go-to for creating dynamic, user-focused interfaces, and I enjoy working with its ecosystem to deliver efficient, scalable front-end solutions.

TypeScript Experience

I have substantial experience with TypeScript, which I've used to enhance the robustness and maintainability of my code. At Innovative Solutions, I implemented a logging system using TypeScript alongside AWS services, which helped improve performance and reduce access times significantly.

I really appreciate how TypeScript adds type safety to JavaScript, making it easier to catch errors during development rather than at runtime. It's particularly helpful in larger projects, like when I was developing APIs or complex applications where keeping track of data types can get tricky.

Additionally, I've integrated TypeScript into React projects to benefit from its features while building components, which has improved both the development experience and the overall code quality. I enjoy the added structure it provides and how it can lead to more reliable and maintainable codebases.

Redux experience

I have solid experience with Redux, especially in managing complex state in React applications. At Ignite-Reading, I utilized Redux to handle global state management, which streamlined data flow across components and improved the overall performance of our application. I implemented actions, reducers, and middleware to manage side effects effectively. In addition to Redux, I've worked with other state management tools like Context API for smaller projects where a full Redux setup might have been overkill. I've also explored Zustand and Recoil for more recent projects, as they offer simpler APIs for managing state in React. Each tool has its strengths, and I like to choose the right one based on the project's needs and complexity. I'm always open to trying new libraries to find the best solutions for state management challenges.

Tech challenges

I've encountered several tech challenges in backend development. One notable challenge was at CMORQ, where I was tasked with deploying a new blockchain application using AWS. We faced issues with scalability due to the high volume of transactions, which led to performance bottlenecks.

To tackle this, I optimized our architecture by implementing a microservices approach and leveraging AWS services like Lambda for serverless computing and DynamoDB for efficient data storage. This significantly improved our app's responsiveness and allowed it to handle increased loads without crashing.

Collaborate with frontend and backend developers to integrate smart contract functionalities experience

Working with front-end and back-end developers to integrate smart contract functionalities has been one of my favorite parts of the job. When we start a project, I usually sit down with the front-end team to understand their needs and how they envision the user interactions. We talk about what kind of data the smart contract will handle and what functions we need to expose.

On the back-end side, I coordinate with developers to make sure the APIs are set up correctly to communicate with the smart contracts. This usually involves defining the endpoints that will

trigger contract functions and return data. I love how everyone brings their expertise to the table, and it's all about finding the best way to connect everything seamlessly.

During integration, I'm often debugging issues together with the teams. If something doesn't work as expected, we'll all hop on a call, go through the code, and troubleshoot the problems together. I find that collaboration really speeds things up and leads to better solutions since we're all aligned on the project goals.

Once everything is integrated, I help with testing to ensure that the front end is interacting correctly with the smart contract. It's exciting to see the whole system come together, especially when we can visualize the smart contract actions in the UI. That collaborative energy makes the process enjoyable and ensures we deliver a smooth user experience!

Perform code reviews, testing, and debugging of smart contracts

Sure! In my experience with code reviews, testing, and debugging smart contracts, I focus on ensuring quality and security. During code reviews, I dive into the logic of the contract, checking for vulnerabilities and efficiency. For example, I once spotted an unnecessary function call that could've increased gas fees, so I suggested a more streamlined approach that saved costs.

When it comes to testing, I write unit tests to cover various scenarios and edge cases, using frameworks that integrate well with Rust and Solana. If issues arise, I dig into transaction logs and recreate the problem locally to troubleshoot effectively. For instance, I encountered a bug in a lending contract that caused unexpected results during transactions, and by collaborating with my teammates, we identified and fixed the logic error, ensuring everything ran smoothly on the mainnet.

Optimize smart contracts for performance and cost efficiency

Optimizing smart contracts for performance and cost efficiency has been a key part of my experience. I always start by analyzing the contract's logic and identifying any areas that could lead to high gas fees. For instance, in a previous project, I noticed that some functions were performing redundant calculations that could be simplified. By refactoring those sections, I managed to reduce gas consumption significantly.

I also pay attention to data storage in contracts, as this can really impact costs. Using data types effectively and minimizing state changes are crucial. In one case, I switched from using multiple storage variables to a single struct, which not only streamlined the code but also reduced the overall transaction cost. By continuously profiling and testing the contract during development, I ensure that we're delivering a solution that's both efficient and cost-effective for users.

Stay updated with the latest developments in blockchain technology and the Solana ecosystem

Staying updated with the latest developments in blockchain technology and the Solana ecosystem is super important to me. I make it a point to follow key industry news through blogs, podcasts, and forums where developers share insights and experiences. I often check out Solana's official documentation and community channels to keep an eye on any updates or new features they're rolling out.

I also enjoy participating in online forums and attending meetups or webinars related to blockchain. For example, I recently joined a Solana developer workshop where we explored new tools and best practices for optimizing smart contracts. Engaging with other developers and exchanging knowledge helps me stay on the cutting edge and ensures that I can apply the latest trends and technologies in my projects. It's exciting to be part of such a rapidly evolving field!

Solid understanding of the Solana blockchain and its architecture.

I have a solid understanding of the Solana blockchain and its architecture, which I've built through hands-on experience and research. Solana's unique proof-of-history consensus mechanism really sets it apart, allowing for high throughput and low latency. This architecture enables me to design smart contracts that can handle a large volume of transactions efficiently, which is essential in the DeFi space.

I'm also familiar with how Solana manages its account model, which differs from traditional blockchains. For instance, understanding the distinction between on-chain accounts and program accounts has helped me optimize data storage in my smart contracts. Overall, my experience with Solana allows me to leverage its strengths effectively while building scalable and secure applications.

Experience with smart contract development, deployment, and maintenance.

I've got a solid background in smart contract development, deployment, and maintenance. When I develop smart contracts, I typically use Rust on the Solana blockchain, focusing on creating secure and efficient code. For instance, in a recent project, I built a lending protocol that involved complex logic for collateral management and interest calculations. I made sure to include thorough testing to catch any potential issues before deployment.

Once the contract is ready, I handle the deployment process, often starting on Devnet to test everything out before moving to the mainnet. After deployment, I focus on maintenance, which includes monitoring for any issues and implementing updates as needed. In one case, I had to upgrade a contract to add new features while ensuring backward compatibility, which taught me a lot about managing contract upgrades safely. Overall, I enjoy the full lifecycle of smart contracts, from development to maintenance!

Familiarity with blockchain protocols, cryptography, and decentralized finance (DeFi) concepts.

I'm quite familiar with blockchain protocols, cryptography, and DeFi concepts, which are essential in my work as a developer. I've worked with various blockchain protocols, including Ethereum and Solana, giving me a good grasp of their unique features and use cases. Understanding how consensus mechanisms work, like proof-of-stake and proof-of-history, helps me choose the right tools for specific projects.

On the cryptography front, I know the basics of hashing, digital signatures, and public-private key infrastructure, which are crucial for ensuring security in decentralized applications. For example, in a recent DeFi project, I implemented a feature that required secure key management for user wallets, ensuring that sensitive data remained protected. Overall, my knowledge of these concepts allows me to build robust and secure solutions in the rapidly evolving DeFi landscape.

Strong problem-solving skills and the ability to work in a fast-paced environment.

I pride myself on my strong problem-solving skills, especially in fast-paced environments. When challenges arise, I like to tackle them head-on, whether it's debugging a tricky smart contract or finding ways to optimize a deployment process. For instance, during a recent project, we faced unexpected latency issues during transaction processing. I quickly analyzed the code and identified a bottleneck in the logic, which allowed us to implement a fix and improve performance significantly.

Working in a dynamic setting is where I thrive. I've collaborated with cross-functional teams, and I enjoy the collaborative energy that comes from brainstorming solutions together. I find that being adaptable and staying calm under pressure helps me contribute effectively to the team's success, whether it's meeting tight deadlines or pivoting on a project based on new requirements. It keeps things exciting and constantly pushes me to grow as a developer!

Excellent communication and teamwork skills.

I really value excellent communication and teamwork skills, as they've been crucial in my career as a developer. I make it a point to keep communication clear and open with my teammates, whether we're discussing project requirements, sharing feedback, or brainstorming solutions. For example, during a recent project, I facilitated daily stand-ups to ensure everyone was on the same page and to address any roadblocks quickly.

Collaboration is key in the tech world, especially when working on complex systems like smart contracts. I love bouncing ideas off my colleagues and learning from their perspectives. By fostering a supportive environment, we've been able to tackle challenges more effectively and deliver high-quality results. I believe that strong teamwork not only enhances productivity but also makes the whole process more enjoyable!

Familiarity with version control systems such as Git.

I'm quite familiar with version control systems, particularly Git, and I use it extensively in my projects. Git has been a game-changer for managing code, especially when working in teams. I rely on it for tracking changes, branching for new features, and collaborating with others without stepping on each other's toes. For example, when I was working on a DeFi application, I created separate branches for different features, which allowed us to work concurrently without conflicts.

I'm also comfortable using tools like GitHub for code reviews and managing pull requests. It's a great way to get feedback from teammates and ensure code quality before merging changes. Overall, my familiarity with Git helps me maintain organized and efficient workflows, making it easier to roll back changes if needed and ensuring that we can collaborate smoothly on complex projects.

Experience with other blockchain platforms and smart contract languages like solidity, Ethereum

I've had some experience with other blockchain platforms and smart contract languages, particularly Ethereum and Solidity. While my primary focus has been on Solana, I've worked on a few projects involving Ethereum smart contracts, where I used Solidity to develop decentralized applications. This gave me a solid understanding of the Ethereum Virtual Machine (EVM) and how different functionalities can be implemented.

One project that stands out was creating a token and a simple decentralized exchange on Ethereum. I got hands-on with features like ERC-20 token standards and interacted with other DeFi protocols using Web3.js. This experience not only enhanced my skills in smart contract development but also broadened my perspective on the different ecosystems in the blockchain space. I appreciate the nuances of each platform, and it's been valuable in shaping my approach to building decentralized applications.

=====

Tell me about yourself

I've been working as a Senior Full Stack Developer for about 8 years now, and I really enjoy diving deep into backend technologies, solving complex problems, and building scalable systems. My background includes leading teams and projects for companies like Revolut and Capgemini, and I recently worked on a personal project where I built a crypto casino game on Solana. I'm always passionate about finding innovative solutions and love the challenge of creating something that's both efficient and robust. Outside of coding, I'm really into gaming and enjoy exploring new tech trends like blockchain and AI. That's pretty much me.

Tell me about your latest role

In my latest role at Ignite-Reading, I worked on the Spark project, which is designed to help admins and tutors manage schools and students more efficiently.

The project had some pages built with Ruby views, but due to performance issues, they needed to be converted to React. So, I refactored the Ruby views to React and collaborated closely with backend teams to integrate APIs. We also developed other projects with similar UI styles, which required reusable components. To address this, I created a private GitHub package for sharing these common components across projects.

Additionally, I led code reviews to maintain high code quality and mentored junior developers. Notably, this was my first experience with Ruby on Rails, but I quickly adapted and built the Ruby backend for the updated features. It was also my first time publishing a private GitHub package, which I successfully completed.

Tell me about your Revolut experience

At Revolut, I was part of the engineering team working on highly complex, scalable solutions. I focused on building and optimizing web applications, making sure they were robust and could handle large-scale traffic. One of the key challenges there was balancing performance with rapid feature development, so I had to stay sharp with both backend and frontend technologies. It was a fast-paced environment, but I really enjoyed working alongside some brilliant minds and pushing my skills to the next level.

Did you review our company?

Yeah, I did some research! I know Playson has been around since 2012 and has built a solid reputation as a leading online gaming supplier. You're focused on creating innovative gaming platforms and promotional features, which really caught my attention. I love that the company emphasizes performance, scalability, and cutting-edge solutions, and it's clear you're passionate about delivering high-quality experiences. The team's "PoC first" approach and sense of ownership are definitely values I align with, and I'm excited to potentially contribute to that.

Why join us?

I'm really excited about the opportunity to join Playson because it seems like a place where I can make a real impact. The focus on performance, scalability, and innovation aligns perfectly with what I love to do. Plus, I enjoy working in environments where ownership is key, and it sounds like Playson encourages that kind of mindset. The gaming industry is always evolving, and the chance to work on promotional features at scale is super exciting to me. Plus, the flexibility, remote work options, and emphasis on professional development are big bonuses!

What are you looking for in your next role?

In my next role, I'm looking for a few key things. First, I want the opportunity to work on innovative projects that challenge me and allow me to grow my technical skills, especially with emerging technologies.

I'm also keen on being part of a collaborative team where knowledge sharing and mentorship are encouraged. A strong engineering culture is important to me—somewhere that values best practices, code quality, and continuous improvement.

Additionally, I appreciate a flexible work environment that supports work-life balance, allowing me to be productive while also pursuing my interests outside of work. Overall, I'm excited about a role that offers both professional growth and a positive team dynamic!

What are your strengths and weaknesses?

****Strength:**** One of my key strengths is my adaptability. I can quickly learn new technologies and frameworks, which has allowed me to successfully transition between various tech stacks and project requirements. This adaptability also extends to collaborating with different teams, as I can effectively communicate and bridge the gap between technical and non-technical stakeholders.

****Weakness:**** As for a weakness, I tend to be detail-oriented, which sometimes leads me to spend more time on a task than necessary. While this ensures high-quality work, I've been working on balancing my attention to detail with the need to meet deadlines. I've started setting clearer priorities and time limits for tasks to help manage this better.

Node.js Experience

I've been working with Node.js for over 5 years now, and it's become my go-to for backend development. I've built everything from APIs to full-scale systems using Node, with a strong focus on performance and scalability. One of my recent projects was a crypto casino game where I used Node.js to handle real-time gameplay and transactions. I've also worked with distributed systems, leveraging Node's asynchronous capabilities to ensure smooth, non-blocking processes. I enjoy how flexible it is and how well it integrates with other technologies like TypeScript and cloud services.

TypeScript experience

I've been using TypeScript for the past few years, especially in projects where scalability and maintainability are important. It's been a game changer for me, providing that extra layer of safety with static typing, which helps catch bugs early. I've used it extensively in backend development with Node.js, and it's great for building complex systems, especially when collaborating with larger teams. The type system really helps make the codebase more

predictable and easier to refactor. Plus, since I work a lot with APIs, having TypeScript ensures smoother integration and fewer issues down the line.

Kafka, RabbitMQ experience

I've worked with both Kafka and RabbitMQ in various projects. Kafka has been my go-to for handling real-time data streams, especially when working with distributed systems. It's great for high-throughput scenarios where you need to process a lot of data reliably. I've used RabbitMQ in cases where message delivery guarantees were crucial, like task queues or microservices communication. Both tools have their strengths, and I've found them super useful for decoupling services and ensuring smooth, asynchronous communication between different parts of a system.

Distributed system experience

I've had substantial experience working with distributed systems, particularly in my previous roles where scalability and resilience were key. I've designed and implemented systems that can handle large volumes of requests across multiple services, which often involved using technologies like Node.js, Kafka, and microservices architecture. One of the challenges I enjoyed tackling was ensuring data consistency and fault tolerance while maintaining performance. I appreciate the complexity of distributed systems and love finding innovative solutions to optimize them. Overall, it's rewarding to build systems that can scale seamlessly and handle failures gracefully.

MySQL, PostgreSQL experience

I've worked extensively with both MySQL and PostgreSQL in my roles at various companies. At Campgemini, I used MySQL for several projects, focusing on optimizing queries and ensuring efficient database performance under heavy loads. I appreciate its reliability and ease of use, especially for applications that require quick access to structured data.

On the other hand, I've used PostgreSQL for projects that needed advanced features like JSONB support and complex queries. Its robust functionality made it ideal for applications at Revolut, where we were building scalable systems with a focus on data integrity. I enjoy leveraging the strengths of each database to meet specific project needs and optimize overall performance.

MongoDB Experience

I've worked with MongoDB in projects that required flexibility and scalability, especially when dealing with unstructured data. During my time at Cognizant Softvision, I used MongoDB to build applications where fast data access and dynamic schemas were crucial. Its document-based model made it easy to iterate quickly, which was essential for our agile development process. I really appreciate how well it integrates with Node.js, allowing us to create efficient data pipelines. Overall, MongoDB has been a great choice for projects that need to scale rapidly and adapt to changing requirements.

E2E test experience

I have solid experience with end-to-end (e2e) testing from my work at Revolut, where we used Node.js for our backend. We implemented e2e tests using tools like Cypress, which allowed us to simulate user interactions effectively and ensure our applications functioned correctly. I focused on writing comprehensive test cases that covered critical user flows, helping us catch issues early in the development process.

Integrating e2e tests into our CI/CD pipeline was essential for maintaining confidence in our releases, and I really enjoyed the challenge of creating robust tests that accurately represented real user behavior. This approach significantly contributed to delivering high-quality features to our users.

Unit Test experience

I have extensive experience with unit testing from my time at Revolut, where we prioritized writing tests to ensure code quality and maintainability. We primarily used tools like Jest for our Node.js applications, allowing us to create isolated tests for individual functions and components.

At Revolut, I focused on developing a robust test suite that covered critical business logic, making it easier to refactor and enhance features without introducing bugs. Writing unit tests became a fundamental part of my workflow, helping to catch issues early and ensuring that each piece of code worked as intended before moving to integration testing. I believe that thorough unit testing is key to delivering reliable and scalable applications.

Large scale microservice architecture experience

I've had significant experience working with large-scale microservices architecture during my time at Revolut. We adopted this approach to ensure that our applications were highly scalable, resilient, and easy to maintain. I was involved in designing and implementing various microservices that communicated through RESTful APIs and message brokers like Kafka.

One of my key responsibilities was to ensure that each microservice was independently deployable, which helped us accelerate our development cycles and improve team autonomy. I also focused on monitoring and optimizing the performance of these services, using tools like Prometheus and Grafana to gain insights into system health and bottlenecks. This experience taught me the importance of service cohesion and loose coupling, which are essential for maintaining a robust architecture as the system grows.

AWS experience

I have hands-on experience with AWS during my time at Revolut, where I utilized services like AWS Lambda, Amplify, Cognito, and S3. I leveraged AWS Lambda for building serverless

functions, which allowed us to handle various tasks efficiently without managing server infrastructure.

Using AWS Amplify, I was able to streamline the development of our applications, enabling rapid deployment and hosting. I also implemented AWS Cognito for user authentication and access control, ensuring a secure experience for our users. Additionally, I used S3 for storing and serving static assets, which was crucial for maintaining performance and scalability. This experience has provided me with a solid foundation in cloud architecture and best practices for building modern applications.

Gambling experience

I have hands-on experience in the gambling domain through a personal project where I developed a crypto casino game on the Solana network. I built the entire application from scratch, using React for the frontend and Node.js with Express for the backend. I also utilized MongoDB for data storage, ensuring efficient management of user data and game state.

The integration of Solana's web3 library was crucial for enabling seamless interactions with the blockchain, including wallet connections and transaction handling. This project allowed me to explore the intricacies of gaming mechanics, user engagement, and secure transactions in a crypto environment. It was a rewarding experience that deepened my understanding of both the gambling industry and blockchain technology.

SaaS Experience

I have experience working with SaaS applications, particularly during my time at Revolut. I was involved in developing scalable web applications that served multiple users while maintaining high availability and performance.

In this role, I focused on building features that catered to various customer needs, ensuring a smooth user experience. I also emphasized implementing security best practices and data privacy measures, which are critical in a SaaS environment. My work included setting up CI/CD pipelines for seamless deployment, allowing us to release updates quickly and efficiently. Overall, my experience has equipped me with a strong understanding of the challenges and best practices involved in building and maintaining SaaS solutions.

Tech challenges you faced at your latest role?

One of the most significant challenges I faced in my latest role at Ignite-Reading was refactoring a large codebase from Ruby on Rails to React. This task involved migrating complex business logic and ensuring that the new React components integrated seamlessly with existing systems. It required careful planning and extensive testing to maintain functionality and performance. Also Ruby on Rails was the first time for me. Additionally, I had to manage the transition without disrupting ongoing development and ensure that the new setup met high standards for maintainability and scalability. Overcoming this challenge not only enhanced the application's

performance but also provided valuable experience in managing large-scale codebase migrations.

How did you resolve this challenge?

To resolve the challenges, I used the TDD (Test-Driven Development) approach. I first wrote tests for the components, ensuring that the core functionality was covered. After that, I generated the necessary React code based on those tests, which helped me catch any potential issues early on. This approach made the migration from Ruby on Rails to React smoother and ensured that I didn't break anything in the overall system. It allowed me to confidently refactor the codebase while maintaining the application's functionality and performance.

Most tech challenges you faced and how you solved it.

In my previous role at Revolut, I faced a significant technical challenge while developing a high-performance crypto trading platform. The platform needed to provide a seamless and responsive user experience for trading and portfolio management, but we encountered major performance bottlenecks during peak trading times.

The platform's responsiveness was compromised, especially when handling real-time data feeds and executing live transactions. The main issues stemmed from inefficient state management and suboptimal API interactions, which led to slow load times and delayed updates. This was critical since, in the trading environment, even minor delays can lead to significant user dissatisfaction and potential financial losses.

I began by thoroughly analyzing the existing state management setup and API interaction patterns. I identified that the platform was using traditional RESTful APIs for data fetching, which resulted in multiple, redundant calls and unnecessary data processing on the client side.

To resolve this, I proposed and implemented a shift from RESTful APIs to GraphQL. By adopting GraphQL, we were able to fetch only the necessary data, reducing the number of API calls and minimizing the amount of data transferred between the client and server.

I refactored the API interaction logic to batch requests and handle them more efficiently, further reducing the server load and enhancing the responsiveness of the application.

These changes resulted in a 30% increase in platform responsiveness and a significant improvement in user engagement. The optimized performance not only enhanced the user experience but also reduced the infrastructure costs associated with handling large volumes of data in real time.

How did you manage the state for this challenge?

I first refactored the state structure to ensure components only subscribed to necessary state slices, which minimized unnecessary re-renders. I transitioned to Redux Toolkit for a more

efficient state management setup, reducing boilerplate code and aligning with best practices. I also implemented memoization using "useEffect", "useMemo" and etc to prevent redundant calculations and re-renders. To complement these changes, I improved our API interaction by batching requests and updating state more efficiently with the new GraphQL setup. Finally, I used performance monitoring tools to continually assess and refine our state management approach.

Challenges you face in collaboration and how did you resolve that?

One challenge I faced in collaboration was aligning on technical decisions with team members who had different opinions. To resolve this, I facilitated open discussions where everyone could present their perspectives and evidence supporting their views. We used a collaborative approach to weigh the pros and cons of each option, and we often used data or user feedback to guide our decisions. This process helped us reach a consensus and ensured that all voices were heard while keeping the project on track.

how do you stay updated in latest tech trends

I stay updated on the latest tech trends by actively following industry blogs, podcasts, and online communities. I regularly read articles on platforms like Medium, Dev.to, and tech-specific newsletters to keep up with new frameworks and best practices. Additionally, I participate in online courses and attend webinars to deepen my knowledge on emerging technologies.

I also engage with developer communities on platforms like GitHub and Stack Overflow, where I can learn from peers and share insights. Attending local meetups and conferences, when possible, allows me to network with other professionals and stay informed about the latest developments in the tech landscape.

=====

Tell me about yourself (looks good than before-fullstack)

I'm Vlad, a Senior Full Stack Developer with about 8 years of experience. I've spent most of my career working on web applications, focusing on building efficient and scalable systems. I'm all about using tech like **React**, **Next.js**, and **Node.js** to solve complex problems and make things work better for users.

I've worked at places like Revolut and Ignite-Reading, leading teams, refactoring legacy systems, and driving improvements with a strong focus on best practices. But beyond the code, I really enjoy mentoring and creating an environment where the team can thrive. I'm always looking to learn new things and push the boundaries on how we can improve processes.

That's pretty much me!

Tell me about your latest role

In my latest role at Ignite-Reading, I worked on the Spark project, which is designed to help admins and tutors manage schools and students more efficiently.

The project had some pages built with Ruby views, but due to performance issues, they needed to be converted to React. So, I refactored the Ruby views to React and collaborated closely with backend teams to integrate APIs. We also developed other projects with similar UI styles, which required reusable components. To address this, I created a private GitHub package for sharing these common components across projects.

Additionally, I led code reviews to maintain high code quality and mentored junior developers. Notably, this was my first experience with Ruby on Rails, but I quickly adapted and built the Ruby backend for the updated features. It was also my first time publishing a private GitHub package, which I successfully completed. Although I'm not yet familiar with machine learning and Python, I am confident in my ability to learn quickly on the job and adapt to new technologies.

Revolut Experience

At Revolut, I worked as part of their web development team, building and maintaining features for their trading platform. The stack there was primarily React on the frontend and Node.js on the backend. My focus was on developing user-facing features, improving performance, and ensuring the security of the application—critical in the fintech space.

I collaborated closely with other teams, like product and design, to make sure the features were aligned with business goals. I also got involved in handling some of the more complex integrations, like connecting third-party APIs for financial services, and I had a chance to help out with code reviews and mentor some junior devs along the way. It was a fast-paced environment, but it definitely sharpened my skills!

1. Trading Functionality

- **Overview:** I developed the trading functionality that allowed users to buy and sell cryptocurrencies on the platform. This feature involved live order execution, integrating real-time data feeds, and handling user transactions efficiently.
- **Technologies:**
 - **Frontend:** React and Next.js for the UI, ensuring a smooth and responsive interface for trading. I used **React Query** for handling data fetching and state management related to the order book and trading data.
 - **Backend:** Used **Node.js** and **AWS Lambda** for serverless functions to handle trade execution, ensuring minimal latency. This architecture allowed scaling without the need for server management.

- **WebSocket:** Implemented **WebSocket** connections to stream real-time order book updates and trade execution statuses. This enabled users to see live updates as they happened.
 - **Security:** Implemented user authentication and trade authorization using **AWS Cognito**.
 - **Functionality:** The system supported market, limit, and stop orders, and executed trades based on real-time data. Transactions were logged, and users received instant feedback on trade status, ensuring transparency.
-

2. GraphQL API for Efficient Data Fetching

- **Overview:** To improve the efficiency of data retrieval, I replaced the existing REST API calls with **GraphQL**. GraphQL allowed clients to request only the data they needed, significantly improving performance.
 - **Technologies:**
 - **Frontend:** **Apollo Client** in the React app to query data using GraphQL. This made it easier to fetch specific data without over-fetching.
 - **Backend:** Used **Node.js** with **Apollo Server** to implement the GraphQL API. **GraphQL resolvers** were optimized for various data points like user portfolios, transaction history, and market data.
 - **Database:** **PostgreSQL** was the primary database, and I optimized the GraphQL queries to pull data efficiently using pagination and batching.
 - **Benefits:** This reduced the number of API calls, improved load times, and allowed for faster and more flexible data interactions. Users could fetch portfolio details, transaction history, or real-time market updates without unnecessary data overhead.
-

3. Portfolio Management Dashboard

- **Overview:** I developed the portfolio management dashboard where users could view and track their crypto assets, including their holdings, market value, and historical performance.
- **Technologies:**
 - **Frontend:** Built using **React**, with **Chart.js** for visualizing the user's portfolio over time. **React Query** was used to handle data fetching and caching, ensuring real-time portfolio updates.
 - **Backend:** The backend was powered by **Node.js**, fetching data from **AWS S3** (for storing historical performance) and **PostgreSQL** for current user balances.
 - **GraphQL:** The dashboard used **GraphQL** queries to fetch portfolio details and transaction history efficiently.

- **State Management:** **Redux** was used for managing the global state of the portfolio across the application, ensuring data consistency and synchronization between different components.
 - **Functionality:** The dashboard displayed crypto holdings, total portfolio value, and profit/loss data. It allowed users to view the performance of individual assets, check historical trends, and manage their assets.
-

4. Real-Time Market Data Integration

- **Overview:** I integrated real-time crypto market data, which was essential for live trading and portfolio monitoring. This involved connecting the platform to external market data providers and ensuring seamless updates across the application.
- **Technologies:**
 - **WebSocket:** I used **WebSocket** connections to stream live market data (e.g., price changes, order books, and trade volumes). This ensured that users always had the most current information when making trades or tracking the market.
 - **Frontend:** In the React app, I utilized **React Query** to manage data updates and ensure that real-time data was reflected instantly on the frontend.
 - **Backend:** The backend was powered by **Node.js** with connections to third-party market data providers, such as **Coinbase API** and **Binance API**, to fetch live market data.
 - **Caching:** Implemented a caching layer using **Redis** to store frequently requested market data, reducing the load on external APIs and speeding up response times.
- **Functionality:** Users could see real-time price changes, track multiple cryptocurrencies simultaneously, and view historical trends. This integration powered not only the trading functionality but also the portfolio and market analysis tools.

AlterCall Experience

At AlterCall AI, I was focused on developing AI-driven solutions for the platform. My role involved integrating AI models into the product, which enhanced the user experience with features like personalized content and automation. I worked primarily with React and Node.js, handling both frontend and backend tasks.

I also got a chance to experiment with some cool AI technologies, helping to bring machine learning capabilities into real-world applications. On top of that, I collaborated with cross-functional teams to implement features and ensure the platform was scalable and efficient. It was a great experience working with cutting-edge tech!

Did you review our company?

Yes, I have reviewed your company and I am very interested. What stands out most to me is how ClaimSorted leverages a mix of AI and human expertise to streamline the entire claims process. The focus on faster claim handling, reducing fraud, and providing real-time data insights while maintaining a high-quality customer experience is particularly impressive. I'm also intrigued by the level of customization you offer through your claims portal and e-FNOL system, allowing clients to tailor the process to their needs. This combination of technology, efficiency, and customer-centric solutions aligns with my passion for building scalable and user-focused platforms.

Why join us?

I'm really excited about NineTwoThree Studio because of the chance to learn new technologies and work on innovative projects. The flexible working hours also stand out to me; I appreciate environments that allow for a good work-life balance.

I particularly like how streamlined your hiring process is—less time in interviews means more time to focus on coding, which is where I feel most comfortable. I'll admit, I'm not the most talkative person, so handling interviews can be a bit challenging for me. But when it comes to coding, I'm right at home! I'm looking forward to the opportunity to contribute my skills and grow with the team in such a supportive environment.

What are you looking for in your next role?

In my next role, I'm looking for a few key things. First, I want the opportunity to work on innovative projects that challenge me and allow me to grow my technical skills, especially with emerging technologies.

I'm also keen on being part of a collaborative team where knowledge sharing and mentorship are encouraged. A strong engineering culture is important to me—somewhere that values best practices, code quality, and continuous improvement.

Additionally, I appreciate a flexible work environment that supports work-life balance, allowing me to be productive while also pursuing my interests outside of work. Overall, I'm excited about a role that offers both professional growth and a positive team dynamic!

What are your strengths and weaknesses?

****Strength:**** One of my key strengths is my adaptability. I can quickly learn new technologies and frameworks, which has allowed me to successfully transition between various tech stacks and project requirements. This adaptability also extends to collaborating with different teams, as I can effectively communicate and bridge the gap between technical and non-technical stakeholders.

****Weakness:**** As for a weakness, I tend to be detail-oriented, which sometimes leads me to spend more time on a task than necessary. While this ensures high-quality work, I've been working on balancing my attention to detail with the need to meet deadlines. I've started setting clearer priorities and time limits for tasks to help manage this better.

Architect experience

I've had experience in architecting solutions, especially in the context of designing scalable and efficient systems. While I haven't officially held the title of "Architect," I've led projects where I designed the overall architecture, particularly in API development and microservices. I've had to make decisions on structuring applications, choosing the right tech stack, and ensuring scalability and maintainability.

For example, in my previous roles, I was responsible for breaking down complex problems into manageable components, designing database schemas, and handling integrations with third-party APIs. It's a natural progression from my full-stack development experience where I've always had to think about how different systems communicate and scale effectively.

Clear and thorough understanding of design patterns and software development best practices

Absolutely! I have a solid grasp of design patterns and software development best practices. I frequently apply patterns like MVC, Singleton, and Observer in my projects, depending on the use case. They help in creating more maintainable and scalable code.

I also emphasize best practices like clean code principles, code reviews, and documentation. Using version control effectively, implementing CI/CD pipelines, and ensuring comprehensive testing are all part of my routine. I believe that following these practices not only enhances code quality but also makes collaboration within the team smoother. It's all about creating a codebase that's easy to understand and maintain over time!

Ability to communicate issues clearly to business stakeholders

I've had plenty of experience communicating technical issues to business stakeholders. In my previous roles, I often acted as a bridge between the development team and non-technical stakeholders.

For example, when we encountered performance issues on a project, I'd explain the technical challenges in layman's terms, highlighting how they impacted user experience and business goals. I'd also provide potential solutions, detailing the pros and cons of each, so stakeholders could make informed decisions.

I believe clear communication is crucial for aligning expectations and ensuring everyone is on the same page, especially when it comes to timelines, project scope, and resource needs. It's all about building that trust and understanding between tech and business sides!

Breaking down complex problems into manageable tasks experience

I've developed a knack for breaking down complex problems into manageable tasks throughout my career. When faced with a large project, I start by analyzing the requirements and identifying the core components.

For example, in my last role, when we needed to revamp a significant feature, I mapped out the user journey and then broke it down into smaller, actionable tasks for the team. Each task had clear objectives and deadlines, which made it easier to track progress and adjust as needed.

I also prioritize tasks based on their impact and dependencies, ensuring that we tackle the most critical issues first. This approach not only keeps the team focused but also helps in maintaining momentum throughout the project. Plus, it encourages collaboration, as everyone can see how their work fits into the bigger picture!

Ability to maintain code style and standards and hold people accountable;

In my experience, maintaining coding standards and holding the team accountable is essential for delivering quality software. At Ignite-Reading, I led code reviews where I emphasized adhering to our established coding guidelines and best practices.

I always approach these discussions constructively, focusing on how improvements can enhance our code quality and maintainability. If someone was falling behind on standards, I'd have a one-on-one chat to understand their challenges and provide support or resources to help them get back on track.

Additionally, I believe in fostering a culture of accountability by setting clear expectations from the start, whether it's around deadlines, coding standards, or project deliverables. When everyone knows what's expected, it creates an environment where team members take ownership of their work and feel empowered to contribute to the project's success.

Microservices architecture experience

I have hands-on experience with microservices architecture in several projects. For instance, I worked on a platform where we transitioned from a monolithic structure to a microservices-based approach. This involved breaking down the application into smaller, independent services that could be developed, deployed, and scaled independently.

I was involved in defining the boundaries of each microservice based on functionality, ensuring clear API contracts for communication between services. I also handled the implementation of service discovery and API gateways to manage the interactions between microservices effectively.

Additionally, I focused on setting up monitoring and logging for each service, which helped us quickly identify issues and improve performance. This experience not only taught me the technical aspects of microservices but also the importance of designing for scalability and resilience, which are crucial in today's applications.

React experience

I've been working with React for several years, primarily focusing on building dynamic and responsive user interfaces. My experience includes developing single-page applications (SPAs) where performance and user experience are critical.

I'm comfortable with React hooks, context API for state management, and integrating it with backend services through RESTful APIs. In my last role, I converted Ruby views to React, which significantly improved the performance of our educational platform.

I've also worked with libraries like Redux for state management and implemented routing using React Router to handle navigation. Beyond just coding, I prioritize writing clean, maintainable code and following best practices, such as component reusability and effective testing using tools like Jest and React Testing Library. Overall, I'm passionate about leveraging React to create seamless user experiences!

Node experience

I have extensive experience with Node.js, using it primarily for backend development. I've built RESTful APIs and microservices that handle various functionalities, such as user authentication, data processing, and third-party integrations.

In my previous roles, I focused on creating efficient and scalable server-side applications, utilizing frameworks like Express.js to streamline development. I've worked with various databases, including PostgreSQL and MongoDB, integrating them with Node.js to manage data effectively.

I also have experience implementing middleware for handling requests, logging, and error management, which are crucial for maintaining application stability. Additionally, I've set up automated testing for Node.js applications using tools like Mocha and Chai, ensuring that the code is reliable and meets the requirements. Overall, Node.js has been a key part of my tech stack, and I enjoy leveraging its capabilities to build robust backend systems!

JavaScript/TypeScript experience

I have a solid foundation in both JavaScript and TypeScript, having worked extensively with them in various projects.

With JavaScript, I've used it for everything from building dynamic web applications to implementing complex logic in both frontend and backend environments. I'm comfortable with

ES6+ features, such as `async/await`, destructuring, and modules, which help make the code more concise and maintainable.

In recent projects, I've shifted towards using TypeScript for its type safety and improved tooling. I find that it helps catch errors during development, making it easier to manage larger codebases. I've integrated TypeScript in React applications, ensuring that components and services have defined types, which significantly enhances code quality and developer experience.

Whether I'm working on state management, API integrations, or building utility functions, I prioritize writing clean, efficient, and maintainable code in both JavaScript and TypeScript. Overall, I appreciate the flexibility of JavaScript and the robustness that TypeScript adds to my development workflow!

React Native experience

I have experience with React Native, primarily working on mobile applications that require a seamless user experience across both iOS and Android platforms. My involvement included developing reusable components and implementing navigation using React Navigation.

In one of my projects, I helped create a mobile app that integrated with a backend API to manage user data and perform real-time updates. I utilized various React Native libraries to enhance functionality, such as handling forms and animations, which improved the overall user interface.

I also focused on optimizing performance by implementing best practices like lazy loading and minimizing unnecessary re-renders. While I've worked on smaller projects with React Native, I'm eager to dive deeper and expand my skills, especially in areas like native module integration and performance optimization. Overall, I enjoy the flexibility React Native offers for building cross-platform apps!

MySQL, PostgreSQL Experience

I have significant experience working with both MySQL and PostgreSQL, using them in various projects to manage relational data effectively.

With **MySQL**, I've designed and optimized database schemas, created stored procedures, and executed complex queries to support backend functionalities. I've also focused on indexing strategies to improve query performance, especially for applications with large datasets.

In terms of **PostgreSQL**, I appreciate its advanced features like JSONB support and full-text search capabilities. I've utilized these features to build efficient and scalable applications, particularly when handling unstructured data. I've also worked on data migrations, ensuring data integrity and performance during transitions.

In both databases, I've implemented CRUD operations through ORM tools like Sequelize and ActiveRecord, which streamlined database interactions. Overall, I'm comfortable managing database connections, optimizing performance, and ensuring data consistency across applications.

MongoDB experience

I have hands-on experience with MongoDB, primarily using it for applications that require flexible data modeling and scalability. I've utilized it in projects where the schema needed to be dynamic, allowing for easy adjustments as the application evolved.

In my work, I've designed and implemented various collections to store related data, leveraging MongoDB's document-based structure. I've also worked with Mongoose, an ODM library, to define models and manage relationships between data, which made it easier to interact with the database.

I've focused on optimizing queries and using indexing strategies to improve performance, especially for applications with large volumes of data. Additionally, I've integrated MongoDB with Node.js applications, enabling seamless data retrieval and manipulation through RESTful APIs.

Overall, I appreciate MongoDB's flexibility and performance capabilities, especially in projects that require rapid development and iteration.

RESTful API design and implementation experience

I have extensive experience in designing and implementing RESTful APIs across various projects. My approach typically involves defining clear and consistent endpoints that align with the principles of REST, ensuring they are intuitive and follow best practices.

In my recent roles, I've worked on creating APIs that support CRUD operations for different resources. I focus on structuring the API responses in a way that's easy for front-end developers to consume, often using JSON format. I also pay attention to proper HTTP methods (GET, POST, PUT, DELETE) to convey the intended action clearly.

For example, I've designed an API for a financial application that handled user accounts, transactions, and reporting features. I implemented authentication using JWTs for secure access and incorporated pagination and filtering in responses to enhance performance and usability.

Additionally, I prioritize writing comprehensive API documentation using tools like Swagger, which helps both developers and stakeholders understand how to interact with the API. Overall, I'm committed to creating APIs that are not only functional but also easy to maintain and extend.

OAuth and single sign-on(SAML)

I have experience implementing OAuth and Single Sign-On (SSO) solutions, particularly with OAuth 2.0 for authorization. In my previous projects, I integrated third-party authentication providers like Google and Facebook to allow users to log in seamlessly using their existing accounts. This not only improves user experience but also enhances security by leveraging trusted identity providers.

I've set up the OAuth flow, managing access tokens and refresh tokens to maintain secure sessions while ensuring proper scopes for data access. Additionally, I focused on handling user permissions and maintaining security best practices throughout the implementation.

Regarding SAML, I've worked on projects where we needed to enable SSO for enterprise applications. I've configured SAML assertions and set up service provider settings to facilitate secure authentication. This included working with identity providers to manage user identities and ensure that the SSO process met security requirements.

Overall, my experience with both OAuth and SAML has given me a solid understanding of authentication and authorization mechanisms, enabling me to create secure and user-friendly applications.

Github experience

I have extensive experience using Git and GitHub for version control and collaboration on software projects. I've worked with Git commands for branching, merging, and managing pull requests to ensure smooth collaboration among team members.

In my previous roles, I often used GitHub for code reviews, where I would submit pull requests for my changes and review others' code to maintain quality and consistency. I appreciate the ability to comment on specific lines and suggest changes, which fosters productive discussions and helps improve code quality.

Additionally, I've published private GitHub packages for internal use, which streamlined dependency management and encouraged code reuse across projects. I also used GitHub Actions for continuous integration and deployment (CI/CD), automating tests and deployments to ensure that our code was always in a deployable state.

Overall, GitHub has been a crucial part of my workflow, enabling effective collaboration, version control, and automation in the development process.

Websocket experience

I have experience using WebSockets to implement real-time communication in applications. In one of my projects, I utilized WebSockets to enable live chat functionality, allowing users to send and receive messages instantly without needing to refresh the page.

I set up a WebSocket server using Node.js, handling connection events, message broadcasting, and managing user sessions. This included ensuring that messages were routed correctly and implementing features like typing indicators and read receipts to enhance user experience. Additionally, I've integrated WebSockets with React to create a responsive frontend that updates in real time based on user interactions. This technology has proven invaluable for applications that require low latency and real-time data updates, such as gaming, notifications, or collaborative tools.

Overall, my experience with WebSockets has taught me how to manage state and performance in real-time applications effectively, providing users with a seamless experience.

Testing library experience

I have experience using various testing libraries to ensure the quality and reliability of the applications I work on.

For **React applications**, I primarily use **Jest** along with **React Testing Library**. This combination allows me to write unit and integration tests that focus on the behavior of components rather than their implementation details. I've written tests to cover component rendering, state changes, and user interactions, which helps catch issues early in the development process.

In addition to React, I've also worked with **Mocha** and **Chai** for testing Node.js applications. These tools enable me to write clear and concise tests for backend functionalities, ensuring that APIs behave as expected and that edge cases are handled appropriately.

I believe in maintaining a robust testing suite to improve code quality and reduce the likelihood of bugs reaching production. Overall, my experience with these testing libraries has helped me foster a culture of quality and reliability in the projects I've worked on.

Scrum experience

I have solid experience working in Scrum environments, participating in various roles throughout the process. In my previous positions, I've been part of cross-functional teams that followed Agile methodologies, which included regular sprint planning, daily stand-ups, sprint reviews, and retrospectives.

During sprint planning, I contributed to defining the sprint goals and breaking down user stories into actionable tasks. I appreciate how this collaborative approach helps align the team on priorities and ensures that everyone is on the same page.

In daily stand-ups, I find it helpful to share updates on what I accomplished, what I'm working on, and any blockers I might have. This transparency fosters communication and allows the team to address issues promptly.

I also value sprint reviews as an opportunity to demonstrate our work to stakeholders and gather feedback, which is crucial for continuous improvement. In retrospectives, I've actively participated in discussing what went well and what can be improved, helping to refine our processes and enhance team dynamics.

Overall, my experience in Scrum has reinforced my belief in the importance of collaboration, adaptability, and continuous improvement in delivering successful projects.

Did you build the web from scratch to deploy?

Yes, I've built web applications from scratch to deployment. In my roles, I've started projects by designing the architecture, setting up the frontend and backend frameworks, and implementing the necessary features. For example, at Revolut, I developed a crypto trading platform, where I handled everything from creating the user interface with React to building the backend with Node.js.

I've also managed the deployment process, ensuring the applications are hosted securely and efficiently, often using services like AWS. My experience includes setting up CI/CD pipelines for automated deployment, monitoring performance, and making iterative improvements based on user feedback.

Why did you leave the last 2 companies?

At Revolut, I completed my project and felt it was time to move on to a role where I could further develop my expertise and contribute to more innovative projects and found Ignite-Reading, an EdTech startup located at US which offers opportunity to make a significant impact on the company success as well as competitive salary. So I ended my role at Revolut.

At Ignite-Reading, it was a contract-based role, and my contract ended in July 2024. Now I'm looking for my next opportunity to further develop my expertise and contribute to innovative projects.

Startup environment you're thinking of?

I think the startup environment to be both exciting and rewarding. The fast-paced, dynamic nature of startups allows me to tackle a variety of challenges and contribute to projects in meaningful ways. I enjoy the opportunity to work closely with a small, dedicated team and have a direct impact on the company's growth and success.

Startups often provide a collaborative atmosphere where innovation is encouraged, and I appreciate the chance to take on diverse responsibilities and learn new skills. While the lack of structure can be challenging, I thrive in environments where I can adapt quickly and help shape the direction of projects. Overall, the energy and potential of a startup are aspects I find very motivating and fulfilling.

Startup companies work experience before?

Yes, I have experience working with startup companies, specifically at Ignite-Reading and AlterCall AI. At both companies, I was involved in a variety of roles and responsibilities due to the fast-paced and dynamic nature of startups. This included developing core features, managing deployments, and adapting to rapidly changing requirements. Working in these

environments taught me to be agile and resourceful, collaborate closely with small teams, and handle diverse challenges effectively. These experiences have strengthened my problem-solving skills and ability to thrive in innovative and high-growth settings.

Startup's advantages and disadvantages?

I find working at a startup really exciting because I get to dive into cool projects and see the impact of my work. I learn a ton quickly and face all kinds of challenges. On the downside, things can be pretty unpredictable and resources might be tight. Demos can be especially stressful with tight deadlines and heavy workloads. But when I nail a demo, it's incredibly satisfying and makes all the hard work worth it. Overall it's a great experience to work in a dynamic environment and enjoy shaping something from scratch

=====

Tell me about yourself

I have around 8 years of experience as a Full-Stack Developer, specializing in JavaScript/TypeScript and frameworks like React, Next.js, and Node, along with both SQL and NoSQL databases. I've consistently focused on building scalable, high-performance web applications and improving code maintainability. Most recently, I worked at Ignite-Reading, a US-based startup, where I refactored key parts of the project to enhance both frontend and backend performance, resolving bottlenecks. I enjoy fast-paced environments and collaborating with teams to deliver efficient, high-quality solutions.

Tell me about your latest role

In my latest role at Ignite-Reading, I worked on the Spark project, which is designed to help admins and tutors manage schools and students more efficiently.

The project had some pages built with Ruby views, but due to performance issues, they needed to be converted to React. So, I refactored the Ruby views to React and collaborated closely with backend teams to integrate APIs. We also developed other projects with similar UI styles, which required reusable components. To address this, I created a private GitHub package for sharing these common components across projects.

Additionally, I led code reviews to maintain high code quality and mentored junior developers. Notably, this was my first experience with Ruby on Rails, but I quickly adapted and built the Ruby backend for the updated features. It was also my first time publishing a private GitHub package, which I successfully completed. Although I'm not yet familiar with machine learning and Python, I am confident in my ability to learn quickly on the job and adapt to new technologies.

Tell me about your role at Revolut

At Revolut, I worked as a Senior Full Stack Developer on their trading platform. My role involved building and optimizing user-facing features with React while managing backend services using Node.js. I worked on integrating third-party APIs, ensuring that the platform remained scalable and secure. Additionally, I refactored several parts of the codebase to improve performance and maintainability, collaborating closely with cross-functional teams to deliver high-quality solutions for the platform.

Tell me about your role at AlterCall

At AlterCall, I worked as a Full Stack Developer, using React for the frontend and Node.js for the backend. We built two projects: a coaching platform and a virtual coaching platform. When I joined, I noticed that the code quality needed improvement, so my first focus was optimizing both the frontend and backend, improving state management, creating reusable components, and reducing redundant requests. I also led the frontend team in developing user-friendly UIs based on Figma designs and regularly conducted code reviews to enforce best practices. This resulted in significant improvements in both code quality and development speed. You can see the platform at altercall.com.

Did you review us?

Yeah, I checked out Tallium! I really like your focus on delivering innovative solutions for industries like fintech, e-commerce, and education. I'm impressed by your commitment to helping businesses grow and your strong emphasis on UX/UI. Plus, the company culture sounds awesome, especially with the team-building activities and social events!

Why join us?

I'm excited about the mission—helping startups thrive financially is impactful work. Plus, I love the idea of working with a passionate, driven team and contributing to something that's making a real difference. The flexibility and remote-first approach are big bonuses too. It just seems like a place where I can grow and make a real impact.

Software Engineering experience

I have around 8 years of experience in software engineering, working across frontend and backend development. I've built and delivered complex web applications using technologies like React, Next.js, Node, and AWS. I've led teams, improved processes, and contributed to successful project outcomes at companies like Ignite-Reading, Revolut, and Capgemini. My focus is always on creating scalable, efficient solutions tailored to business needs.

Front End development experience

I have extensive experience in frontend development, primarily using React and Next.js. I've built responsive and high-performance user interfaces, worked with libraries like Material-UI,

and created reusable components to streamline development. I'm skilled in optimizing performance, integrating APIs, and using state management tools like Redux. I've also worked with design systems and tools like Storybook to improve collaboration and consistency across teams.

Backend development experience

I've worked extensively on backend development using Node.js (with frameworks like Nest.js and Express), as well as Ruby on Rails. I've built and maintained RESTful APIs, worked with GraphQL for efficient data fetching, and handled complex integrations with third-party systems. I have experience with databases like PostgreSQL, MongoDB, and MySQL, and I've used AWS services like Lambda, Cognito, and S3 to build scalable and secure backend systems. My focus is always on creating reliable, high-performance backend solutions that meet business needs.

React experience.

I've been working with React for about 8 years now, and it's been a major part of my tech stack. I've developed user-friendly interfaces for various projects, building reusable components and implementing state management solutions to boost performance. My experience also includes optimizing rendering for speed, integrating APIs, and using tools like Redux and Zustand for state management. I write tests using Jest, Vitest, React Testing Library, and Storybook to ensure quality. I'm also dedicated to maintaining high code quality through regular code reviews and best practices. Most recently, I published a private React library to share common components across multiple projects.

Node experience

If you ask me to pick my top tech stacks, I would definitely choose React and Node.js. Like with React, I focus on Node.js to develop backend APIs for around 8 years. I've developed RESTful APIs and integrated various third-party services to enhance application functionality. My experience includes working with frameworks like Express.js and Nest.js, where I've implemented middleware for authentication and data validation. I've also optimized database queries with SQL and NoSQL databases to ensure efficient data handling. Additionally, I prioritize writing clean, maintainable code and have led code reviews to enforce best practices within the team.

TypeScript experience

I've been working with TypeScript for several years, mainly in frontend projects with React and backend projects using Node.js. I like how it adds type safety and helps catch errors early. I've used it to build scalable and maintainable applications, create type definitions for complex APIs, and enhance code quality across large projects. It's definitely my go-to for JavaScript development.

SQL experience

I have extensive experience working with SQL databases, particularly MySQL and PostgreSQL. I've used these databases to design and optimize data models for various applications, ensuring data integrity and performance. My experience includes writing complex queries, optimizing indexing, and managing database migrations. I also focus on efficient data retrieval and manipulation, implementing best practices for query performance. Additionally, I've integrated SQL databases with backend services in Node.js, ensuring seamless data handling and storage for applications.

MongoDB experience

I've used MongoDB in several projects, mainly for building scalable and flexible databases. I like its schema-less nature, which makes it great for rapid development. I've implemented data models, handled CRUD operations, and optimized queries for performance. Additionally, I've integrated MongoDB with Node.js applications, ensuring efficient data retrieval and storage. It's been a key part of my stack for handling unstructured data and real-time applications.

AWS experience

I have significant experience with AWS from my roles at Revolut and AlterCall AI. At Revolut, I utilized services like Lambda for serverless functions, S3 for scalable storage, and Cognito for secure user authentication. At AlterCall AI, I used AWS for hosting applications, managing databases, and ensuring high availability. My focus has always been on optimizing performance and cost-efficiency while ensuring security and scalability across the applications I've developed.

integrate applications with Android and Apple platforms experience

I don't have direct experience integrating applications with Android and Apple platforms, but I've worked on web applications that prioritize responsive design and can be adapted for mobile use. I understand the importance of optimizing user interfaces for different devices and ensuring smooth functionality across platforms.

For instance, I've built applications with React that are mobile-friendly and can be easily transformed into progressive web apps (PWAs). I also have experience collaborating with mobile development teams, so I'm familiar with the challenges and considerations involved in integrating web solutions with mobile platforms. I'm eager to learn more about this area and apply my skills in future projects.

Financial platform experience

I have experience working on financial platforms, particularly at Revolut, where I engineered a high-performance crypto trading platform using React and Next.js. I focused on creating a seamless user experience for trading and portfolio management, integrating real-time data feeds, and optimizing performance for scalability.

I also utilized AWS services for secure user authentication and data storage, which helped ensure reliability and security. My work involved close collaboration with cross-functional teams to integrate backend services and ensure a robust trading experience.

While I haven't worked directly on traditional finance software, my experience in fintech has given me valuable insights into building solutions that cater to the financial sector's unique needs.

Did you build the web from scratch to deploy?

Yes, I've built web applications from scratch to deployment. In my roles, I've started projects by designing the architecture, setting up the frontend and backend frameworks, and implementing the necessary features. For example, at Revolut, I developed a crypto trading platform, where I handled everything from creating the user interface with React to building the backend with Node.js.

I've also managed the deployment process, ensuring the applications are hosted securely and efficiently, often using services like AWS. My experience includes setting up CI/CD pipelines for automated deployment, monitoring performance, and making iterative improvements based on user feedback.

Why did you leave the last 2 companies?

At Revolut, I completed my project and felt it was time to move on to a role where I could further develop my expertise and contribute to more innovative projects and found Ignite-Reading, an EdTech startup located at US which offers opportunity to make a significant impact on the company success as well as competitive salary. So I ended my role at Revolut.

At Ignite-Reading, it was a contract-based role, and my contract ended in July 2024. Now I'm looking for my next opportunity to further develop my expertise and contribute to innovative projects.

Startup environment you're thinking of?

I think the startup environment to be both exciting and rewarding. The fast-paced, dynamic nature of startups allows me to tackle a variety of challenges and contribute to projects in meaningful ways. I enjoy the opportunity to work closely with a small, dedicated team and have a direct impact on the company's growth and success.

Startups often provide a collaborative atmosphere where innovation is encouraged, and I appreciate the chance to take on diverse responsibilities and learn new skills. While the lack of structure can be challenging, I thrive in environments where I can adapt quickly and help shape the direction of projects. Overall, the energy and potential of a startup are aspects I find very motivating and fulfilling.

Startup companies work experience before?

Yes, I have experience working with startup companies, specifically at Ignite-Reading and AlterCall AI. At both companies, I was involved in a variety of roles and responsibilities due to the fast-paced and dynamic nature of startups. This included developing core features, managing deployments, and adapting to rapidly changing requirements. Working in these environments taught me to be agile and resourceful, collaborate closely with small teams, and handle diverse challenges effectively. These experiences have strengthened my problem-solving skills and ability to thrive in innovative and high-growth settings.

Startup's advantages and disadvantages?

I find working at a startup really exciting because I get to dive into cool projects and see the impact of my work. I learn a ton quickly and face all kinds of challenges. On the downside, things can be pretty unpredictable and resources might be tight. Demos can be especially stressful with tight deadlines and heavy workloads. But when I nail a demo, it's incredibly satisfying and makes all the hard work worth it. Overall it's a great experience to work in a dynamic environment and enjoy shaping something from scratch

Tech challenges you faced at your latest role?

One of the most significant challenges I faced in my latest role at Ignite-Reading was refactoring a large codebase from Ruby on Rails to React. This task involved migrating complex business logic and ensuring that the new React components integrated seamlessly with existing systems. It required careful planning and extensive testing to maintain functionality and performance. Also Ruby on Rails was the first time for me. Additionally, I had to manage the transition without disrupting ongoing development and ensure that the new setup met high standards for maintainability and scalability. Overcoming this challenge not only enhanced the application's performance but also provided valuable experience in managing large-scale codebase migrations.

How did you resolve this challenge?

To resolve the challenges, I used the TDD (Test-Driven Development) approach. I first wrote tests for the components, ensuring that the core functionality was covered. After that, I generated the necessary React code based on those tests, which helped me catch any potential issues early on. This approach made the migration from Ruby on Rails to React smoother and ensured that I didn't break anything in the overall system. It allowed me to confidently refactor the codebase while maintaining the application's functionality and performance.

Most tech challenges you faced and how you solved it.

In my previous role at Revolut, I faced a significant technical challenge while developing a high-performance crypto trading platform. The platform needed to provide a seamless and

responsive user experience for trading and portfolio management, but we encountered major performance bottlenecks during peak trading times.

The platform's responsiveness was compromised, especially when handling real-time data feeds and executing live transactions. The main issues stemmed from inefficient state management and suboptimal API interactions, which led to slow load times and delayed updates. This was critical since, in the trading environment, even minor delays can lead to significant user dissatisfaction and potential financial losses.

I began by thoroughly analyzing the existing state management setup and API interaction patterns. I identified that the platform was using traditional RESTful APIs for data fetching, which resulted in multiple, redundant calls and unnecessary data processing on the client side.

To resolve this, I proposed and implemented a shift from RESTful APIs to GraphQL. By adopting GraphQL, we were able to fetch only the necessary data, reducing the number of API calls and minimizing the amount of data transferred between the client and server.

I refactored the API interaction logic to batch requests and handle them more efficiently, further reducing the server load and enhancing the responsiveness of the application.

These changes resulted in a 30% increase in platform responsiveness and a significant improvement in user engagement. The optimized performance not only enhanced the user experience but also reduced the infrastructure costs associated with handling large volumes of data in real time.

How did you manage the state for this challenge?

I first refactored the state structure to ensure components only subscribed to necessary state slices, which minimized unnecessary re-renders. I transitioned to Redux Toolkit for a more efficient state management setup, reducing boilerplate code and aligning with best practices. I also implemented memoization using "useEffect", "useMemo" and etc to prevent redundant calculations and re-renders. To complement these changes, I improved our API interaction by batching requests and updating state more efficiently with the new GraphQL setup. Finally, I used performance monitoring tools to continually assess and refine our state management approach.

Challenges you face in collaboration and how did you resolve that?

One challenge I faced in collaboration was aligning on technical decisions with team members who had different opinions. To resolve this, I facilitated open discussions where everyone could present their perspectives and evidence supporting their views. We used a collaborative approach to weigh the pros and cons of each option, and we often used data or user feedback to guide our decisions. This process helped us reach a consensus and ensured that all voices were heard while keeping the project on track.

how do you stay updated in latest tech trends

I stay updated on the latest tech trends by actively following industry blogs, podcasts, and online communities. I regularly read articles on platforms like Medium, Dev.to, and tech-specific newsletters to keep up with new frameworks and best practices. Additionally, I participate in online courses and attend webinars to deepen my knowledge on emerging technologies.

I also engage with developer communities on platforms like GitHub and Stack Overflow, where I can learn from peers and share insights. Attending local meetups and conferences, when possible, allows me to network with other professionals and stay informed about the latest developments in the tech landscape.

=====

Tell me about yourself

I have around 8 years of experience as a Full-Stack Developer, specializing in JavaScript/TypeScript and frameworks like React, Next.js, and Node, along with both SQL and NoSQL databases. I've consistently focused on building scalable, high-performance web applications and improving code maintainability. Most recently, I worked at Ignite-Reading, a US-based startup, where I refactored key parts of the project to enhance both frontend and backend performance, resolving bottlenecks. I enjoy fast-paced environments and collaborating with teams to deliver efficient, high-quality solutions.

Tell me about your latest role

In my latest role at Ignite-Reading, I worked on the Spark project, which is designed to help admins and tutors manage schools and students more efficiently.

The project had some pages built with Ruby views, but due to performance issues, they needed to be converted to React. So, I refactored the Ruby views to React and collaborated closely with backend teams to integrate APIs. We also developed other projects with similar UI styles, which required reusable components. To address this, I created a private GitHub package for sharing these common components across projects.

Additionally, I led code reviews to maintain high code quality and mentored junior developers. Notably, this was my first experience with Ruby on Rails, but I quickly adapted and built the Ruby backend for the updated features. It was also my first time publishing a private GitHub package, which I successfully completed. Although I'm not yet familiar with machine learning and Python, I am confident in my ability to learn quickly on the job and adapt to new technologies.

Tell me about your role at Revolut

At Revolut, I worked as a Senior Full Stack Developer on their trading platform. My role involved building and optimizing user-facing features with React while managing backend services using Node.js. I worked on integrating third-party APIs, ensuring that the platform remained scalable and secure. Additionally, I refactored several parts of the codebase to improve performance and maintainability, collaborating closely with cross-functional teams to deliver high-quality solutions for the platform.

Tell me about your role at AlterCall

At AlterCall, I worked as a Full Stack Developer, using React for the frontend and Node.js for the backend. We built two projects: a coaching platform and a virtual coaching platform. When I joined, I noticed that the code quality needed improvement, so my first focus was optimizing both the frontend and backend, improving state management, creating reusable components, and reducing redundant requests. I also led the frontend team in developing user-friendly UIs based on Figma designs and regularly conducted code reviews to enforce best practices. This resulted in significant improvements in both code quality and development speed. You can see the platform at altercall.com.

Did you review us?

Yeah, I checked you out! You're doing some great work helping startups manage their finances and avoid cash flow issues. I saw you've been around for a few years, bootstrapped and profitable, and just recently raised \$1M in Seed Funding. Love the mission to support early-stage businesses and boost GDP growth. It's cool that you're working with companies like Reface and Petcube too!

Why join us?

I'm excited about the mission—helping startups thrive financially is impactful work. Plus, I love the idea of working with a passionate, driven team and contributing to something that's making a real difference. The flexibility and remote-first approach are big bonuses too. It just seems like a place where I can grow and make a real impact.

Startup environment you're thinking of?

I think the startup environment to be both exciting and rewarding. The fast-paced, dynamic nature of startups allows me to tackle a variety of challenges and contribute to projects in meaningful ways. I enjoy the opportunity to work closely with a small, dedicated team and have a direct impact on the company's growth and success.

Startups often provide a collaborative atmosphere where innovation is encouraged, and I appreciate the chance to take on diverse responsibilities and learn new skills. While the lack of structure can be challenging, I thrive in environments where I can adapt quickly and help shape

the direction of projects. Overall, the energy and potential of a startup are aspects I find very motivating and fulfilling.

Startup companies work experience before?

Yes, I have experience working with startup companies, specifically at Ignite-Reading and AlterCall AI. At both companies, I was involved in a variety of roles and responsibilities due to the fast-paced and dynamic nature of startups. This included developing core features, managing deployments, and adapting to rapidly changing requirements. Working in these environments taught me to be agile and resourceful, collaborate closely with small teams, and handle diverse challenges effectively. These experiences have strengthened my problem-solving skills and ability to thrive in innovative and high-growth settings.

Startup's advantages and disadvantages?

I find working at a startup really exciting because I get to dive into cool projects and see the impact of my work. I learn a ton quickly and face all kinds of challenges. On the downside, things can be pretty unpredictable and resources might be tight. Demos can be especially stressful with tight deadlines and heavy workloads. But when I nail a demo, it's incredibly satisfying and makes all the hard work worth it. Overall it's a great experience to work in a dynamic environment and enjoy shaping something from scratch

React experience.

I've been working with React for about 8 years now, and it's been a major part of my tech stack. I've developed user-friendly interfaces for various projects, building reusable components and implementing state management solutions to boost performance. My experience also includes optimizing rendering for speed, integrating APIs, and using tools like Redux and Zustand for state management. I write tests using Jest, Vitest, React Testing Library, and Storybook to ensure quality. I'm also dedicated to maintaining high code quality through regular code reviews and best practices. Most recently, I published a private React library to share common components across multiple projects.

Node experience

If you ask me to pick my top tech stacks, I would definitely choose React and Node.js. Like with React, I focus on Node.js to develop backend APIs for around 8 years. I've developed RESTful APIs and integrated various third-party services to enhance application functionality. My experience includes working with frameworks like Express.js and Nest.js, where I've implemented middleware for authentication and data validation. I've also optimized database queries with SQL and NoSQL databases to ensure efficient data handling. Additionally, I prioritize writing clean, maintainable code and have led code reviews to enforce best practices within the team.

SQL experience

I have extensive experience working with SQL databases, particularly MySQL and PostgreSQL. I've used these databases to design and optimize data models for various applications, ensuring data integrity and performance. My experience includes writing complex queries, optimizing indexing, and managing database migrations. I also focus on efficient data retrieval and manipulation, implementing best practices for query performance. Additionally, I've integrated SQL databases with backend services in Node.js, ensuring seamless data handling and storage for applications.

Application refactoring and performance optimization experience.

I have a strong background in application refactoring and performance optimization, which has been a key focus in my recent roles. For instance, at Ignite-Reading, I refactored parts of the project to resolve performance bottlenecks, enhancing both the frontend and backend. I've optimized rendering in React applications, improved state management, and reduced redundant API calls to boost efficiency.

In previous roles at AlterCall and Revolut, I consistently evaluated and updated code to improve maintainability and performance. This included rewriting inefficient components, streamlining database queries, and implementing caching strategies. My approach involves using profiling tools to identify slow sections of the application and applying best practices to ensure that applications run smoothly and efficiently.

Tech challenges you faced at your latest role?

One of the most significant challenges I faced in my latest role at Ignite-Reading was refactoring a large codebase from Ruby on Rails to React. This task involved migrating complex business logic and ensuring that the new React components integrated seamlessly with existing systems. It required careful planning and extensive testing to maintain functionality and performance. Also Ruby on Rails was the first time for me. Additionally, I had to manage the transition without disrupting ongoing development and ensure that the new setup met high standards for maintainability and scalability. Overcoming this challenge not only enhanced the application's performance but also provided valuable experience in managing large-scale codebase migrations.

How did you resolve this challenge?

To resolve the challenges, I used the TDD (Test-Driven Development) approach. I first wrote tests for the components, ensuring that the core functionality was covered. After that, I generated the necessary React code based on those tests, which helped me catch any potential issues early on. This approach made the migration from Ruby on Rails to React smoother and ensured that I didn't break anything in the overall system. It allowed me to confidently refactor the codebase while maintaining the application's functionality and performance.

Most tech challenges you faced and how you solved it.

In my previous role at Revolut, I faced a significant technical challenge while developing a high-performance crypto trading platform. The platform needed to provide a seamless and responsive user experience for trading and portfolio management, but we encountered major performance bottlenecks during peak trading times.

The platform's responsiveness was compromised, especially when handling real-time data feeds and executing live transactions. The main issues stemmed from inefficient state management and suboptimal API interactions, which led to slow load times and delayed updates. This was critical since, in the trading environment, even minor delays can lead to significant user dissatisfaction and potential financial losses.

I began by thoroughly analyzing the existing state management setup and API interaction patterns. I identified that the platform was using traditional RESTful APIs for data fetching, which resulted in multiple, redundant calls and unnecessary data processing on the client side.

To resolve this, I proposed and implemented a shift from RESTful APIs to GraphQL. By adopting GraphQL, we were able to fetch only the necessary data, reducing the number of API calls and minimizing the amount of data transferred between the client and server.

I refactored the API interaction logic to batch requests and handle them more efficiently, further reducing the server load and enhancing the responsiveness of the application.

These changes resulted in a 30% increase in platform responsiveness and a significant improvement in user engagement. The optimized performance not only enhanced the user experience but also reduced the infrastructure costs associated with handling large volumes of data in real time.

How did you manage the state for this challenge?

I first refactored the state structure to ensure components only subscribed to necessary state slices, which minimized unnecessary re-renders. I transitioned to Redux Toolkit for a more efficient state management setup, reducing boilerplate code and aligning with best practices. I also implemented memoization using "useEffect", "useMemo" and etc to prevent redundant calculations and re-renders. To complement these changes, I improved our API interaction by batching requests and updating state more efficiently with the new GraphQL setup. Finally, I used performance monitoring tools to continually assess and refine our state management approach.

Problem you haveChallenges you face in collaboration and how did you resolve that?

One challenge I faced in collaboration was aligning on technical decisions with team members who had different opinions. To resolve this, I facilitated open discussions where everyone could

present their perspectives and evidence supporting their views. We used a collaborative approach to weigh the pros and cons of each option, and we often used data or user feedback to guide our decisions. This process helped us reach a consensus and ensured that all voices were heard while keeping the project on track.

how do you stay updated in latest tech trends

I stay updated on the latest tech trends by actively following industry blogs, podcasts, and online communities. I regularly read articles on platforms like Medium, Dev.to, and tech-specific newsletters to keep up with new frameworks and best practices. Additionally, I participate in online courses and attend webinars to deepen my knowledge on emerging technologies.

I also engage with developer communities on platforms like GitHub and Stack Overflow, where I can learn from peers and share insights. Attending local meetups and conferences, when possible, allows me to network with other professionals and stay informed about the latest developments in the tech landscape.

how do you maintain code quality?

I maintain code quality through a combination of best practices and collaborative efforts within the team. First, I emphasize writing clean, readable, and well-structured code, adhering to coding standards and guidelines. I use tools like ESLint and Prettier to enforce style consistency and catch potential issues early.

Code reviews are a crucial part of my process; I regularly review my teammates' code and welcome feedback on mine. This not only helps identify bugs and improve code quality but also fosters knowledge sharing within the team.

I also write unit and integration tests using tools like Jest and React Testing Library to ensure that the code functions as expected. Continuous integration tools help automate testing and deployment, allowing us to catch issues before they reach production. Overall, maintaining a culture of quality and accountability is key to delivering reliable software.

What principles do you apply?

I apply SOLID, DRY, and KISS principles to improve code quality. These practices help ensure that my code is well-structured, maintainable, and easy to understand. By following the SOLID principles, I promote better object-oriented design. The DRY principle helps eliminate redundancy, while the KISS principle keeps my code simple and straightforward. This approach leads to cleaner, more efficient code that's easier to test and refactor over time.

Tell me about DRY, KISS and SOLID principles

Sure! So, the DRY principle stands for "Don't Repeat Yourself." It emphasizes the importance of reducing redundancy in your code. Instead of duplicating code, you create reusable functions or

components. This not only makes your codebase cleaner but also easier to maintain since changes need to be made in one place rather than multiple locations.

Then we have the SOLID principles, which are a set of five design principles that help developers create more understandable, flexible, and maintainable software. The "S" stands for Single Responsibility Principle, meaning a class should only have one reason to change. The "O" is for Open/Closed Principle, which states that software entities should be open for extension but closed for modification. The "L" is for Liskov Substitution Principle, suggesting that objects should be replaceable with instances of their subtypes without altering the correctness of the program. The "I" is for Interface Segregation Principle, meaning that no client should be forced to depend on methods it doesn't use. Finally, the "D" stands for Dependency Inversion Principle, which emphasizes that high-level modules should not depend on low-level modules, but both should depend on abstractions.

And then there's KISS, which stands for "Keep It Simple, Stupid." The idea here is to avoid overcomplicating your code. The simpler the code, the easier it is to understand, maintain, and debug. It's all about focusing on simplicity and avoiding unnecessary complexity, which can lead to issues down the line.

Using these principles together helps me write better code that is easier to maintain and extend, ultimately leading to higher quality software.

====Interview with Pragya====

Tell me about yourself

I am TaiXun, a software engineer with about 5 years of experience, specializing in JavaScript and its frameworks like React, Next.js, and Node.js, alongside Python and Django for building high-performance web applications. Throughout my career, I have collaborated with several clients and companies, working on a wide range of tech stacks, which has given me deep expertise in JavaScript/TypeScript and its frameworks. Additionally, I possess foundational knowledge in areas such as Python, AI, Django, and blockchain technologies.

Tell me about your latest role

In my last role, I worked on the Spark project at Ignite-Reading, where we helped admins and tutors manage schools and students better. I mainly refactored some slow Ruby views to React and teamed up with the backend devs to hook up APIs. Since we were building a bunch of projects with similar UIs, I created a private GitHub package to share reusable components across them. I also led code reviews to keep the quality high. Even though it was my first time using Ruby on Rails, I picked it up fast and helped build out the backend. Oh, and it was also my first time publishing a private GitHub package, which went smoothly!

I handled the main tasks assigned to me, while my manager was responsible for daily standups.

I've been working as a software engineer for about 5 years, mostly for clients who are working at companies like Ignite-Reading. I handle the tasks assigned to them and get paid for my work.

React Experience

I've been working with React for most of my time as a developer, mainly building fast, high-performing web apps. I've used it for everything from refactoring big projects to creating reusable components. I've also worked with React Query, Material UI, and Redux for managing state. One cool project I did was creating a private GitHub package to share components across multiple apps, which made life a lot easier for the team. Overall, I'm all about writing clean, maintainable code and working closely with backend teams to make sure the APIs are solid.

Node Experience

I've worked extensively with Node.js throughout my career, focusing on building high-performance backend services and RESTful APIs. Over the years, I've developed everything from simple APIs to complex server-side applications using Node.js with Express.js, and I've integrated tools like GraphQL to optimize data flow between frontend and backend.

At Capgemini, for instance, I was responsible for several Node.js-driven applications where I implemented robust backend services to enhance both performance and reliability, achieving a 30% improvement in response times. I'm also experienced with using Node.js in serverless environments, like AWS Lambda, which I utilized at Revolut to streamline functions and cut down on infrastructure costs.

I tend to approach Node with a focus on scalability and efficiency, using design patterns that suit both the team's development speed and the project's future growth.

Python experience

I've been using Python on and off for a few years, mostly for scrapping and backend work and building web apps with Django. It's been great for writing clean, readable code, especially when working on APIs or data-heavy projects. I've also worked in some AI stuff and machine learning libraries like TensorFlow, but I wouldn't call myself an expert in that area. Mostly, I stick to using Python for automating tasks and creating backend systems that need to handle lots of data efficiently.

=====

Tell me about yourself

I'm Vlad, a Senior Full Stack Developer with around 8 years of experience in web development. I've worked extensively with React and TypeScript, leading projects that focus on creating dynamic, user-friendly interfaces. At my previous role, I was involved in shaping frontend architecture and establishing best practices for the team. I'm passionate about performance optimization and state management, and I enjoy mentoring junior developers to help them grow.

In addition to my frontend skills, I've got a solid background in backend technologies, which gives me a well-rounded perspective when collaborating with product teams. I'm excited about the opportunity at ChurchDesk to leverage my experience and contribute to meaningful projects that help churches modernize and connect with their communities.

Tell me about your latest role.

In my latest role at Ignite-Reading, I worked on the Spark project, which is designed to help admins and tutors manage schools and students more efficiently. Admins could sign in to handle tasks like assigning or removing students from tutors, managing session timeouts, and adding or removing entities. Tutors could manage student data, schedule sessions, and track daily activities.

The project had some pages built with Ruby views, but due to performance issues, they needed to be converted to React. So, I refactored the Ruby views to React and collaborated closely with backend teams to integrate APIs. We also developed other projects with similar UI styles, which required reusable components. To address this, I created a private GitHub package for sharing these common components across projects.

Additionally, I led code reviews to maintain high code quality and mentored junior developers. Notably, this was my first experience with Ruby on Rails, but I quickly adapted and built the Ruby backend for the updated features. It was also my first time publishing a private GitHub package, which I successfully completed. Although I'm not yet familiar with machine learning and Python, I am confident in my ability to learn quickly on the job and adapt to new technologies.

Tell me about your role at Revolut

At Revolut, I played a key role in building the backend for our crypto trading platform. My main focus was on designing and developing scalable backend services that could handle high-volume trading efficiently. I worked extensively with Node.js and Express to develop APIs that supported real-time trading and integrated with various crypto exchanges, ensuring low latency and fast execution, which is critical in trading systems.

On the performance side, I optimized server-side logic to reduce response times and ensure the platform could handle heavy traffic, especially during market spikes. I worked closely with our database team to improve performance, using PostgreSQL and MongoDB to manage trade

data and user transactions, balancing between relational and NoSQL databases based on use case.

I also contributed to blockchain integration. We worked with EVM-based systems to manage wallets, handle deposits and withdrawals, and ensure secure transactions. While blockchain wasn't the primary focus of my role, I implemented smart contracts where needed, ensuring they were secure and efficient.

Additionally, I led a small team of engineers, conducting code reviews, guiding the team through the development process, and ensuring we adhered to best practices for security and scalability. I collaborated with the front-end team to ensure the backend services aligned with the user interface for smooth trading experiences. Overall, my role was a mix of hands-on development, system optimization, and team leadership.

Tell me about your role at AlterCall

At AlterCall AI, I worked as a Full Stack Developer, building scalable web applications. On the front-end, I used React and Material-UI to create clean, responsive interfaces. On the back-end, I developed Node.js APIs and integrated them with the front-end to ensure smooth communication between the client and server. I also utilized AWS for hosting and database management, making sure the apps were reliable and scalable. Mentoring junior developers and conducting code reviews were also part of my role, which helped improve overall code quality and team efficiency..

Did you review our company?

Yes, I checked out ChurchDesk and found that you focus on helping churches modernize and engage with their communities through digital tools. I admire how you've grown from a startup into an international business while still emphasizing support and community connection. The mission resonates with me, and I'm excited about the potential to make a real impact in that space! And I would like to learn more about the company from you.

Why join us?

I'm really drawn to ChurchDesk because of your mission to help churches modernize and connect with their communities. It's a unique space where technology meets meaningful impact, and I love the idea of working on projects that truly make a difference. Plus, the supportive work environment and opportunities for growth align perfectly with what I'm looking for in my next role. I'm excited about the chance to collaborate with talented people and contribute to a product that serves such important institutions!

What are you looking for in your next role?

I'm looking for a role where I can dive into new technologies and work on innovative projects. I want to be part of a team where I can contribute to meaningful solutions and continue to grow

both technically and professionally. I'm also excited about opportunities to mentor and collaborate with others, and I appreciate a role that offers autonomy and the chance to shape the development process.

What is the motivation you work on?

My main motivation is seeing the impact of what I build. I love creating solutions that not only work well but make a real difference for the end user. I'm also driven by learning—whether it's new tech, improving my skills, or mentoring others. For me, it's about constantly growing and delivering something meaningful.

React Experience

I've been working with React for several years, primarily focusing on building dynamic user interfaces. I've developed complex applications that leverage state management libraries like Redux and React Query for efficient data handling. I also prioritize performance optimization, ensuring smooth user experiences by implementing techniques like lazy loading and code splitting.

In my recent projects, I've shaped frontend architecture, established best practices for the team, and mentored junior developers. I enjoy using modern tools and libraries within the React ecosystem to create scalable solutions that enhance user engagement. Overall, I'm passionate about leveraging React to solve real-world problems and create intuitive applications!

State management

State management is all about how to handle and store the data that drives the application's UI. In React, managing state can be done using built-in features like `useState` and `useReducer`, but for larger applications, I often turn to libraries like Redux or React Query.

Redux allows for a centralized store, making it easier to manage global state across components, while React Query simplifies server state management, handling data fetching, caching, and synchronization automatically. I focus on keeping state predictable and easy to debug, which ultimately leads to a better user experience. It's crucial to choose the right approach based on the complexity of the app and its data flow!

State management in React in detail

Managing state in React can be done in several ways, depending on the complexity of your application:

1. **Local State**: For simple components, I can use the `useState` hook to manage local state. It's straightforward and perfect for managing UI states like form inputs or toggle switches.

2. **Context API**: For managing global state without prop, the Context API allows to create a global state provider that can be accessed by any component in the tree.
3. **useReducer Hook**: For more complex state management, `useReducer` can be helpful. It allows you to handle state transitions more predictably, similar to Redux, by using actions and a reducer function.
4. **State Management Libraries**: For larger applications, libraries like Redux or MobX offer more powerful state management solutions. Redux, for instance, centralizes your application's state and provides tools for debugging and time-traveling.
5. **React Query**: When dealing with server state, React Query is great for fetching, caching, and synchronizing data with your server, making it easier to manage API responses.

Choosing the right approach depends on your app's size and complexity, but these options cover a wide range of scenarios!

Redux summary

Redux is a state management library designed to help manage application state in a predictable way, especially in larger applications. It centers around a single store that holds the entire state of the app, making it easier to track and debug. Actions, which are plain objects with a `type` and optional `payload`, describe changes to the state and are dispatched to signal those changes.

Reducers are pure functions that take the current state and an action, returning a new state based on that action. This separation of concerns allows for clear state transitions. Middleware like Redux Thunk can handle asynchronous actions, such as API calls, enhancing the store's capabilities. React-Redux provides bindings to integrate Redux seamlessly with React, allowing components to access the store and dispatch actions easily. Overall, Redux promotes a structured approach to state management, making complex apps more manageable and maintainable.

React-Query summary

TanStack Query, formerly React Query, is a data-fetching library designed to simplify server state management in React applications. It automatically caches fetched data, which reduces unnecessary requests and boosts performance. The library supports background updates, keeping your UI synchronized with the latest data without manual refreshes.

You can easily invalidate and refetch queries based on specific events, and it also offers automatic retries for failed requests, enhancing reliability. TanStack Query simplifies implementing features like pagination and infinite scrolling, making it easier to work with large datasets. Overall, it streamlines data management, allowing developers to focus more on building features rather than handling data fetching complexities.

React Query experience

I've used React Query extensively in my projects to manage server state efficiently. It has significantly simplified data fetching and caching, allowing me to focus on building features rather than handling the complexities of state management.

With React Query, I've implemented background data synchronization, ensuring that the UI stays up-to-date with minimal effort. The automatic caching has improved performance by reducing unnecessary API calls, and I appreciate how easy it is to invalidate and refetch queries when data changes.

Typescript experience

I've been using TypeScript extensively in my projects, especially in conjunction with React. TypeScript's strong typing helps catch errors early, making the code more robust and easier to maintain. I appreciate features like interfaces and enums, which enhance code clarity and facilitate collaboration within the team.

In my recent roles, I've integrated TypeScript into existing JavaScript codebases, improving overall code quality and developer experience. It allows for better tooling support and auto-completion, which speeds up development. I also enjoy leveraging TypeScript to create type-safe APIs and ensure that components receive the right props, reducing runtime errors. Overall, TypeScript has become an essential part of my frontend development toolkit!

Webpack, babel and ESLint experience

I've worked with Webpack for bundling my JavaScript applications, using it to manage assets, optimize performance, and set up various plugins to enhance the development experience. I often configure it to enable code splitting and lazy loading, which helps improve load times.

Babel is another tool I rely on, especially for transpiling modern JavaScript and TypeScript to ensure compatibility across different browsers. I appreciate how it allows me to use the latest features without worrying about browser support.

For linting and code quality, I regularly use ESLint. It helps enforce coding standards and catch potential issues early in the development process. I often customize ESLint configurations to fit the team's preferences and ensure consistent code style across our projects. Together, these tools create a streamlined development workflow that enhances productivity and code quality!

React Native experience

I have some experience with React Native, primarily through a part-time role where I contributed to developing a mobile app. I worked on integrating a multi-chain wallet and adapted code from React Native to create a web app version.

I enjoyed using familiar React concepts, like components and state management, to build a smooth user experience. While I haven't worked extensively with React Native in a professional capacity, I appreciate how it allows for cross-platform development, and I'm eager to deepen my skills in that area, especially as it relates to mobile apps and in-person solutions!

Collaborate with product managers, providing technical insights to help shape and refine product features experience

I've always placed a strong emphasis on collaboration with product managers throughout my projects. At Revolut and AlterCall, I frequently worked alongside product teams to understand user needs and translate them into technical requirements. I provided insights on feasibility, suggested potential features, and helped refine product roadmaps based on technical constraints and opportunities.

By maintaining open communication, we were able to align on priorities and ensure that the development process was smooth. I believe that this collaboration is crucial for delivering a product that not only meets technical standards but also truly serves user needs. I enjoy being an active part of that conversation, bridging the gap between technical and non-technical teams!

Backend integration and RESTful APIs experience.

I have solid experience with backend integration and working with RESTful APIs. In my roles, I've built and consumed APIs to connect the frontend with various backend services, ensuring smooth data flow and user interactions.

I've implemented features that require fetching, updating, and deleting data via RESTful endpoints, handling authentication and error management along the way. My background in both frontend and backend development gives me a well-rounded perspective, allowing me to design APIs that are intuitive and easy for frontend developers to work with. Overall, I see effective API integration as a key part of creating seamless user experiences!

Leading and mentoring others experience

In my previous roles, I've taken on leadership responsibilities by mentoring junior developers and guiding teams through project challenges. I've organized code reviews to help team members improve their skills and encouraged best practices in our development process.

I believe in fostering a collaborative environment where everyone feels comfortable sharing ideas and asking questions. I've also led workshops and training sessions to introduce new technologies or concepts, helping to boost team knowledge and confidence. It's rewarding to see team members grow and succeed, and I'm passionate about creating a culture of learning and support!

=====

Apply practical knowledge of TypeScript and Nest.js in developing software experience

In my recent roles, I've been using TypeScript extensively for building scalable, type-safe applications. I have solid experience with Nest.js, especially from projects where I needed to develop robust APIs and manage complex integrations. I've worked on systems that require strong architecture, and Nest.js has been great for structuring services, handling dependency injection, and keeping the codebase maintainable.

Design robust APIs and scalable backend architectures

I've designed and built robust APIs in several projects, focusing on scalability and performance. At Revolut, I worked on microservices architecture, ensuring APIs could handle high traffic and complex transactions. I also led backend architecture design for fintech applications, making sure it could scale as user bases grew. For example, I used REST and GraphQL for API design and implemented solutions like load balancing and caching to optimize performance.

SQL and non-SQL experience

I have experience working with both SQL and non-SQL databases. For SQL, I've used PostgreSQL and MySQL in projects where data integrity and complex queries were crucial, like handling financial transactions. For non-SQL, I've worked with MongoDB to manage unstructured data and support high-volume applications, particularly in gaming projects. I'm comfortable choosing the right database based on the project's needs and designing schemas that ensure efficiency and scalability.

Deep understanding of ECMAScript specifications

I have a strong grasp of ECMAScript specifications, which has helped me write clean, efficient JavaScript code. I stay updated with the latest features and best practices, like async/await for handling asynchronous operations and ES6 modules for better code organization. This understanding allows me to leverage the full power of JavaScript, optimize performance, and ensure compatibility across different environments.

ECMAScript is the standard that defines the scripting language JavaScript. It outlines the syntax, semantics, and core features that JavaScript implementations must follow. Regular updates introduce new features and improvements, like ES6 (or ES2015), which brought in major additions like classes, modules, and arrow functions. Essentially, ECMAScript ensures consistency across different environments where JavaScript runs, making it easier for developers to write reliable code.

Git experience

I've been using Git for version control in all my projects. I manage branches for features, handle merges, and conduct code reviews through pull requests. It's helped me maintain a clean and organized codebase, collaborate effectively with team members, and track changes over time. I'm also familiar with Git workflows in GitHub for repository management and CI/CD integration.

aws cloud service experience

At Revolut, I leveraged AWS services extensively for deploying and managing applications. I used EC2 for scalable hosting and S3 for secure storage of user data and application assets. I also implemented AWS Lambda to create serverless functions, optimizing performance and reducing costs. At AlterCall, I worked with AWS RDS for managing databases and integrated various services to enhance our app's functionality. This experience has given me a strong foundation in building efficient, cloud-based solutions.

Fintech experience

Yes, I've worked at Revolut, which is a well-known fintech company. There, I focused on developing scalable backend systems and integrating payment solutions. I gained hands-on experience with financial transactions, security protocols, and compliance measures. Additionally, my work involved designing APIs that supported both fiat and cryptocurrency transactions, giving me a solid understanding of the fintech landscape.

scrum experience

In my latest role, I was actively involved in the Scrum process as a Senior Full Stack Developer. I took part in sprint planning meetings to help define and prioritize tasks, participated in daily stand-ups to discuss progress and blockers, and contributed to sprint retrospectives to reflect on what went well and what could be improved. I also worked closely with the Scrum Master and product owner to ensure that the team's work aligned with project goals and deadlines.

Tech challenges you faced at your latest role?

One of the most significant challenges I faced in my latest role at Ignite-Reading was refactoring a large codebase from Ruby on Rails to React. This task involved migrating complex business logic and ensuring that the new React components integrated seamlessly with existing systems. It required careful planning and extensive testing to maintain functionality and performance. Also Ruby on Rails was the first time for me. Additionally, I had to manage the transition without disrupting ongoing development and ensure that the new setup met high standards for maintainability and scalability. Overcoming this challenge not only enhanced the application's performance but also provided valuable experience in managing large-scale codebase migrations.

How did you resolve this challenge?

To resolve the challenges, I used the TDD (Test-Driven Development) approach. I first wrote tests for the components, ensuring that the core functionality was covered. After that, I generated the necessary React code based on those tests, which helped me catch any potential issues early on. This approach made the migration from Ruby on Rails to React smoother and ensured that I didn't break anything in the overall system. It allowed me to confidently refactor the codebase while maintaining the application's functionality and performance.

Most tech challenges you faced and how you solved it.

In my previous role at Revolut, I faced a significant technical challenge while developing a high-performance crypto trading platform. The platform needed to provide a seamless and responsive user experience for trading and portfolio management, but we encountered major performance bottlenecks during peak trading times.

The platform's responsiveness was compromised, especially when handling real-time data feeds and executing live transactions. The main issues stemmed from inefficient state management and suboptimal API interactions, which led to slow load times and delayed updates. This was critical since, in the trading environment, even minor delays can lead to significant user dissatisfaction and potential financial losses.

I began by thoroughly analyzing the existing state management setup and API interaction patterns. I identified that the platform was using traditional RESTful APIs for data fetching, which resulted in multiple, redundant calls and unnecessary data processing on the client side.

To resolve this, I proposed and implemented a shift from RESTful APIs to GraphQL. By adopting GraphQL, we were able to fetch only the necessary data, reducing the number of API calls and minimizing the amount of data transferred between the client and server.

I refactored the API interaction logic to batch requests and handle them more efficiently, further reducing the server load and enhancing the responsiveness of the application.

These changes resulted in a 30% increase in platform responsiveness and a significant improvement in user engagement. The optimized performance not only enhanced the user experience but also reduced the infrastructure costs associated with handling large volumes of data in real time.

How did you manage the state for this challenge?

I first refactored the state structure to ensure components only subscribed to necessary state slices, which minimized unnecessary re-renders. I transitioned to Redux Toolkit for a more efficient state management setup, reducing boilerplate code and aligning with best practices. I also implemented memoization using "useEffect", "useMemo" and etc to prevent redundant calculations and re-renders. To complement these changes, I improved our API interaction by

batching requests and updating state more efficiently with the new GraphQL setup. Finally, I used performance monitoring tools to continually assess and refine our state management approach.

Challenges you face in collaboration and how did you resolve that?

One challenge I faced in collaboration was aligning on technical decisions with team members who had different opinions. To resolve this, I facilitated open discussions where everyone could present their perspectives and evidence supporting their views. We used a collaborative approach to weigh the pros and cons of each option, and we often used data or user feedback to guide our decisions. This process helped us reach a consensus and ensured that all voices were heard while keeping the project on track.

=====

Experience in leading teams and mentoring junior developers

I've had several opportunities to lead teams and mentor junior developers throughout my career. At Revolut, I led a small team of backend engineers, guiding them through the development process for our crypto trading platform. I conducted regular code reviews to ensure quality, shared best practices for performance and security, and helped junior developers improve their coding and problem-solving skills. I also took the time to mentor them on broader concepts like system architecture, scalability, and working with complex integrations. My goal has always been to foster growth while maintaining high standards and delivering projects on time.

did you create web from scratch?

Yes, I've created web applications from scratch. For example, at AlterCall AI, I developed the AlterCall project using React based on Figma designs. I was responsible for turning those designs into a fully functional web app.

Additionally, I worked on a personal project called Pirate Rush, which was a crypto casino game built with React and Node. Although the site is currently down, you can still see how it looked on YouTube. This project involved designing and building the game from scratch, including the integration with blockchain technologies.

give me an example how you optimized the server-side logic

At Revolut, we faced performance issues during high-traffic periods, especially during volatile market conditions. To optimize the server-side logic, I started by profiling our Node.js application to identify bottlenecks, particularly in database queries and API response times.

One example was reducing the overhead of multiple database queries by implementing query batching and caching frequently accessed data. For instance, I optimized the order book API by combining several smaller queries into a single bulk query and cached static data like market pairs. This reduced database load and improved response times significantly.

Additionally, I refactored parts of the code to use asynchronous operations more efficiently with promises and `async/await`, avoiding blocking operations. These changes helped us reduce API response times, ensuring the platform could handle high volumes of transactions without delays.

Blockchain Experience

In my previous roles, I led several blockchain integrations, primarily focusing on EVM-based systems like Ethereum and Binance Smart Chain. At Revolut, I worked on integrating smart contracts for wallet management and secure transactions. This involved writing and deploying Solidity contracts, ensuring they met security standards, and managing blockchain interactions.

For a personal project, I developed a crypto casino game called PirateRush on the Solana network. I used React for the frontend, Node for the backend, and Solana web3 and Metaplex for blockchain interaction. I minted NFTs for presale and built games like mines, coin flip, turtle racing, loot box, and PvP turtle racing. The game is down now, but you can still see it on YouTube.

mobile app development experience

I worked on mobile app development during a part-time role at A&C Tech, where I helped build the BioMatrix mobile app using React Native. I integrated a multi-chain wallet with the app and used AWS Face Rekognition to generate user wallets based on facial recognition. Afterward, I transitioned the app to a web version by converting the React Native code to React, ensuring consistency between both platforms. This experience gave me solid exposure to mobile app development, even though it's not listed on my resume.

Your React Experience

Yes, I have extensive experience with React. I've used it to build complex, dynamic user interfaces and manage state efficiently using tools like Redux and React Query. I've worked on various projects, from creating responsive web apps to integrating APIs and optimizing performance. React's component-based architecture has been a big advantage in keeping my code organized and maintainable.

TypeScript experience

I've been using TypeScript for a while now and really appreciate how it enhances code quality and maintainability. I've used it extensively with React to catch errors early and improve developer productivity. I've worked on converting JavaScript projects to TypeScript to leverage its strong typing features and make the codebase more robust. Overall, TypeScript has been a valuable tool for ensuring consistency and reliability in my projects.

Node experience

I have extensive experience with Node.js. I've used it for building backend services, handling API requests, and managing server-side logic. I've worked with various Node.js frameworks and libraries, such as Express.js and Nest.js, to create scalable and efficient applications. My experience includes working with Node.js for real-time data processing, integrating with databases, and ensuring smooth performance in production environments.

Redux-based state management experience

I've worked with Redux for state management in several projects, particularly for complex state management in React applications. I've used Redux to manage application state, handle complex data flows, and ensure that components are updated consistently. I'm familiar with setting up Redux stores, defining actions and reducers, and using middleware like Redux Thunk for handling asynchronous operations. This experience has helped me maintain predictable state management and improve the overall performance and maintainability of the applications I've worked on.

GraphQL and REST API experience on both developing and integration

I've worked extensively with both GraphQL and REST APIs in my development roles.

****For REST APIs:****

- I've designed and implemented RESTful services using Node.js and Express, focusing on creating scalable endpoints and ensuring efficient data retrieval and manipulation.
- I've handled various integration scenarios, such as connecting frontend applications to backend services and managing authentication and data validation.

****For GraphQL:****

- I've developed GraphQL APIs to provide flexible and efficient data querying. This involved setting up schemas, resolvers, and integrating with databases to support complex queries and relationships.
- I've also worked on integrating GraphQL with frontend frameworks like React to allow for more efficient data fetching and reduce over-fetching or under-fetching issues.

In both cases, I've focused on optimizing performance, ensuring security, and providing a smooth experience for both developers and users.

Solidity experience

I have hands-on experience with Solidity, particularly in developing and deploying smart contracts for blockchain projects. At Revolut, I worked on integrating smart contracts for wallet management and secure transactions. This involved writing Solidity contracts, conducting thorough testing to ensure they were secure and efficient, and deploying them on Ethereum and Binance Smart Chain. I also managed contract interactions, optimized gas usage, and ensured compliance with security best practices to protect against common vulnerabilities.

scrum experience

In my latest role, I was actively involved in the Scrum process as a Senior Full Stack Developer. I took part in sprint planning meetings to help define and prioritize tasks, participated in daily stand-ups to discuss progress and blockers, and contributed to sprint retrospectives to reflect on what went well and what could be improved. I also worked closely with the Scrum Master and product owner to ensure that the team's work aligned with project goals and deadlines.

Your think about startup environement?

I think the startup environment to be both exciting and rewarding. The fast-paced, dynamic nature of startups allows me to tackle a variety of challenges and contribute to projects in meaningful ways. I enjoy the opportunity to work closely with a small, dedicated team and have a direct impact on the company's growth and success.

Startups often provide a collaborative atmosphere where innovation is encouraged, and I appreciate the chance to take on diverse responsibilities and learn new skills. While the lack of structure can be challenging, I thrive in environments where I can adapt quickly and help shape the direction of projects. Overall, the energy and potential of a startup are aspects I find very motivating and fulfilling.

Startup companies work experience before?

Yes, I have experience working with startup companies, specifically at Ignite-Reading and AlterCall AI. At both companies, I was involved in a variety of roles and responsibilities due to the fast-paced and dynamic nature of startups. This included developing core features, managing deployments, and adapting to rapidly changing requirements. Working in these environments taught me to be agile and resourceful, collaborate closely with small teams, and handle diverse challenges effectively. These experiences have strengthened my problem-solving skills and ability to thrive in innovative and high-growth settings.

Startup's advantages and disadvantages?

I find working at a startup really exciting because I get to dive into cool projects and see the impact of my work. I learn a ton quickly and face all kinds of challenges. On the downside, things can be pretty unpredictable and resources might be tight. Demos can be especially stressful with tight deadlines and heavy workloads. But when I nail a demo, it's incredibly satisfying and makes all the hard work worth it. Overall, if I thrive in a dynamic environment and enjoy shaping something from scratch, it's a great experience.

=====

Tell me about JavaScript and asynchronous programming.

JavaScript is a key part of both front-end and back-end development. Asynchronous programming in JavaScript helps manage tasks like API calls or file operations without freezing the main thread. I use callbacks, promises, and `async/await` to handle these tasks. Promises let me manage operations that will finish later, and `async/await` makes writing asynchronous code cleaner and more readable. This approach ensures my applications stay responsive and efficient while handling multiple tasks at once.

Experience with RESTful APIs and integrating third-party services.

I've worked extensively with RESTful APIs, both creating and consuming them to connect front-end and back-end components. At AlterCall AI, I built RESTful APIs to facilitate smooth communication between the client and server. I've also integrated third-party services, like real-time data feeds at Revolut, to enhance functionality and provide users with up-to-date information. This experience has helped me ensure seamless interactions and improve overall application performance.

Tell me about Git and version control systems.

I use Git and version control systems to manage and track changes in my code. Git helps me keep a history of code changes, collaborate with others by managing branches, and handle pull requests. It's essential for tracking revisions, resolving conflicts, and maintaining a smooth workflow in team environments. I rely on it to ensure code quality and to integrate updates efficiently.

What do you think about work collaboratively in a team environment and communicate effectively.

I believe working collaboratively in a team is crucial for delivering high-quality projects. Effective communication helps ensure everyone is on the same page, solves problems quickly, and fosters a supportive environment. I'm all about sharing knowledge, providing constructive feedback, and working together to achieve common goals. It makes the whole process more efficient and enjoyable.

What was the challenges you face at collaboration and how did you resolve that?

One challenge I faced was coordinating across different time zones and ensuring everyone stayed aligned. To resolve this, I implemented regular check-ins and clear documentation to keep everyone informed, regardless of when they could join the discussions. I also used collaboration tools effectively to share updates and track progress. This approach helped keep the team connected and streamlined our workflow.

Tech challenges you faced at your latest role?

One of the most significant challenges I faced in my latest role at Ignite-Reading was refactoring a large codebase from Ruby on Rails to React. This task involved migrating complex business logic and ensuring that the new React components integrated seamlessly with existing systems. It required careful planning and extensive testing to maintain functionality and performance. Also Ruby on Rails was the first time for me. Additionally, I had to manage the transition without disrupting ongoing development and ensure that the new setup met high standards for maintainability and scalability. Overcoming this challenge not only enhanced the application's performance but also provided valuable experience in managing large-scale codebase migrations.

What was the tech challenges you face at node backend development and how did you resolve that?

A challenge I faced was handling real-time data with Node.js in a high-performance trading platform. The system needed to process and deliver real-time updates efficiently without lag. I resolved this by implementing WebSockets for real-time communication and using Redis for caching and message brokering. Additionally, I optimized database queries and ensured efficient handling of concurrent connections to maintain performance and reliability under high load.

In my previous role at Revolut, I faced a significant technical challenge while developing a high-performance crypto trading platform. The platform needed to provide a seamless and responsive user experience for trading and portfolio management, but we encountered major performance bottlenecks during peak trading times.

The platform's responsiveness was compromised, especially when handling real-time data feeds and executing live transactions. The main issues stemmed from inefficient state management and suboptimal API interactions, which led to slow load times and delayed

updates. This was critical since, in the trading environment, even minor delays can lead to significant user dissatisfaction and potential financial losses.

I began by thoroughly analyzing the existing state management setup and API interaction patterns. I identified that the platform was using traditional RESTful APIs for data fetching, which resulted in multiple, redundant calls and unnecessary data processing on the client side.

To resolve this, I proposed and implemented a shift from RESTful APIs to GraphQL. By adopting GraphQL, we were able to fetch only the necessary data, reducing the number of API calls and minimizing the amount of data transferred between the client and server.

I refactored the API interaction logic to batch requests and handle them more efficiently, further reducing the server load and enhancing the responsiveness of the application.

These changes resulted in a 30% increase in platform responsiveness and a significant improvement in user engagement. The optimized performance not only enhanced the user experience but also reduced the infrastructure costs associated with handling large volumes of data in real time.

How did you manage the state for this challenge?

I first refactored the state structure to ensure components only subscribed to necessary state slices, which minimized unnecessary re-renders. I transitioned to Redux Toolkit for a more efficient state management setup, reducing boilerplate code and aligning with best practices. I also implemented memoization using "useEffect", "useMemo" and etc to prevent redundant calculations and re-renders. To complement these changes, I improved our API interaction by batching requests and updating state more efficiently with the new GraphQL setup. Finally, I used performance monitoring tools to continually assess and refine our state management approach.

Difference between RESTful APIs and graphql?

First, REST Apis often require multiple endpoints to fetch related pieces of data while graphql allows clients to specify exactly what they need in a single request.

Also REST apis typically require multiple requests to different endpoints to gather related data while with graphql you can aggregate data from multiple sources in a single query.

=====

Your React Experience

I've got loads of experience with React, from building responsive UIs to modernizing legacy systems. Lately, I've been all about making components scalable and maintaining performance, using tools like Redux and React Query, and typing things out with TypeScript. I also focus on optimizing performance and mentoring junior devs to keep everything up to scratch.

Next.js Experience

I have experience with Next.js, which I've used to build high-performance, server-rendered React applications. In my previous roles, I leveraged Next.js for its server-side rendering capabilities, which improved SEO and initial page load times.

I've worked with Next.js to create dynamic, server-rendered pages and implement static site generation where appropriate. This involved configuring and optimizing Next.js settings, handling API routes, and integrating with various backend services. Additionally, I've utilized Next.js's built-in features for routing, data fetching, and optimizing performance.

This experience has given me a strong understanding of how to use Next.js to build scalable and efficient web applications that deliver a seamless user experience.

React vs next.js

React and Next.js are both powerful tools for building web applications, but they serve different purposes and offer distinct benefits.

React is a JavaScript library for building user interfaces. It focuses on creating reusable components and managing the state of your application. React is excellent for building single-page applications (SPAs) with rich, interactive user experiences. It provides the flexibility to structure your application as you see fit, but you might need to handle routing, server-side rendering, and other aspects separately.

Next.js, on the other hand, is a framework built on top of React that provides additional features and optimizations. It includes server-side rendering (SSR) and static site generation (SSG) out of the box, which can enhance performance and SEO. Next.js also comes with built-in routing, which simplifies the development process. It offers automatic code splitting, optimized performance, and support for API routes, making it a more opinionated and integrated solution compared to React alone.

In summary, while React is a flexible library for building UIs, Next.js adds server-side capabilities and a set of conventions that can streamline development, improve performance, and enhance SEO for your applications.

server-side rendering in Next.js experience with best practice

I've worked a lot with server-side rendering in Next.js. Basically, I've used SSR to make sure pages load quickly and are SEO-friendly by rendering content on the server before sending it to the client. I've also set up caching to keep things fast and reduce server load. I handle dynamic data fetching with `getServerSideProps` and `getStaticProps`, so pages get the right data when needed. Plus, I keep an eye on performance and tweak things as needed to keep everything running smoothly.

Collaborate experience with designers and backend developers to build a seamless user experience

I've collaborated closely with designers and backend developers to create a smooth user experience. With designers, I make sure the frontend aligns with their designs and bring their visions to life using React and other technologies. I'm proactive in discussing user flows and design elements to ensure everything looks and works as intended. When working with backend developers, I focus on integrating APIs and services seamlessly, making sure the data flows smoothly between the frontend and backend. Communication and regular check-ins are key to ensure we're all on the same page and deliver a cohesive, high-quality experience.

AWS experience

I've got solid experience with AWS. I've used AWS Cognito for user authentication, S3 for scalable storage, and Lambda for serverless functions. I'm comfortable setting up and managing these services to ensure applications are secure, scalable, and cost-effective. I've also used various other AWS tools to optimize performance and manage infrastructure, making sure everything runs smoothly and efficiently.

Write clean, maintainable, and testable code

I prioritize writing clean, maintainable, and testable code by following best practices and guidelines. I focus on keeping my code organized and modular, so it's easy to understand and modify. I also use consistent naming conventions and document my code to ensure it's clear for others. For maintainability, I make sure to write reusable components and functions. To ensure quality, I write thorough tests using tools like Jest and Cypress, and I'm diligent about code reviews to catch potential issues early and maintain high standards.

how did you mentored junior developers, what was the challenges and how did you resolve that?

Mentoring junior developers has been a key part of my role. I've guided them through best practices, code reviews, and hands-on coding challenges. One of the main challenges was helping them navigate complex codebases and understand advanced concepts. I resolve this by breaking down tasks into manageable chunks and providing clear, actionable feedback. Regular

one-on-one sessions and pair programming also helped address their specific questions and learning needs. Over time, this approach improved their skills and confidence, and it also fostered a collaborative and supportive team environment.

Tech challenges you faced at your latest role?

One of the most significant challenges I faced in my latest role at Ignite-Reading was refactoring a large codebase from Ruby on Rails to React. This task involved migrating complex business logic and ensuring that the new React components integrated seamlessly with existing systems. It required careful planning and extensive testing to maintain functionality and performance. Also Ruby on Rails was the first time for me. Additionally, I had to manage the transition without disrupting ongoing development and ensure that the new setup met high standards for maintainability and scalability. Overcoming this challenge not only enhanced the application's performance but also provided valuable experience in managing large-scale codebase migrations.

Most tech challenges you faced and how you solved it.

In my previous role at Revolut, I faced a significant technical challenge while developing a high-performance crypto trading platform. The platform needed to provide a seamless and responsive user experience for trading and portfolio management, but we encountered major performance bottlenecks during peak trading times.

The platform's responsiveness was compromised, especially when handling real-time data feeds and executing live transactions. The main issues stemmed from inefficient state management and suboptimal API interactions, which led to slow load times and delayed updates. This was critical since, in the trading environment, even minor delays can lead to significant user dissatisfaction and potential financial losses.

I began by thoroughly analyzing the existing state management setup and API interaction patterns. I identified that the platform was using traditional RESTful APIs for data fetching, which resulted in multiple, redundant calls and unnecessary data processing on the client side.

To resolve this, I proposed and implemented a shift from RESTful APIs to GraphQL. By adopting GraphQL, we were able to fetch only the necessary data, reducing the number of API calls and minimizing the amount of data transferred between the client and server.

I refactored the API interaction logic to batch requests and handle them more efficiently, further reducing the server load and enhancing the responsiveness of the application.

These changes resulted in a 30% increase in platform responsiveness and a significant improvement in user engagement. The optimized performance not only enhanced the user

experience but also reduced the infrastructure costs associated with handling large volumes of data in real time.

How did you manage the state for this challenge?

I first refactored the state structure to ensure components only subscribed to necessary state slices, which minimized unnecessary re-renders. I transitioned to Redux Toolkit for a more efficient state management setup, reducing boilerplate code and aligning with best practices. I also implemented memoization using "useEffect", "useMemo" and etc to prevent redundant calculations and re-renders. To complement these changes, I improved our API interaction by batching requests and updating state more efficiently with the new GraphQL setup. Finally, I used performance monitoring tools to continually assess and refine our state management approach.

Difference between RESTful APIs and graphql?

First, REST Apis often require multiple endpoints to fetch related pieces of data while graphql allows clients to specify exactly what they need in a single request.

Also REST apis typically require multiple requests to different endpoints to gather related data while with graphql you can aggregate data from multiple sources in a single query.

====Meeting with dorte@awork.com 16/09

Tell me about yourself

I'm a Senior Full Stack Developer with extensive experience in frontend technologies like React, Next.js, and TypeScript. Over the past decade, I've led teams, improved processes, and built scalable web applications. I'm passionate about delivering high-quality user experiences and staying current with the latest tech trends. In my previous roles, I've focused on optimizing performance, enhancing user interfaces, and integrating modern tools and workflows. I'm excited about the opportunity to apply my skills and contribute to your team's success as a Senior Frontend Engineer.

Tell me about your latest role

In my latest role as a Senior Software Engineer at Ignite-Reading, I focused on converting key components from Ruby on Rails to React, which significantly improved the system's performance and maintainability. I also published a private GitHub package, which streamlined our development process by making reusable components easily accessible across projects.

In addition to my development work, I led code reviews and mentored junior developers, ensuring they followed best practices and maintained high standards in their work. This helped

elevate the overall quality of our codebase and fostered a culture of continuous improvement within the team.

Why join us?

I'm drawn to your team because you prioritize creating a joyful work environment and tackle challenging projects. Your focus on modern project management tools and continuous improvement aligns with my passion for delivering high-quality software and fostering collaboration. Plus, the flexible work options and team-centric culture sound like a great fit for my skills and work style.

What are you looking for in your next role?

I'm looking for a role where I can dive into challenging projects, work with a talented team, and keep learning. I want to build cool stuff with modern tech, and I'm excited about opportunities to contribute to both the code and the company culture. Flexibility and a collaborative environment are big pluses for me, too.

What is the motivation you work on?

I'm motivated by tackling complex problems and seeing projects come to life. I love the challenge of building scalable solutions, improving user experiences, and collaborating with a great team. The chance to learn new things and make a real impact drives me every day.

Your Angular Experience

I haven't used Angular directly, but I've got a solid background in React and Next.js, so I'm pretty familiar with component-based setups and state management. I'm good at picking up new frameworks quickly and applying what I know to new challenges. I'm confident I can get up to speed with Angular and bring my experience to the table.

Your React Experience

I've got loads of experience with React, from building responsive UIs to modernizing legacy systems. Lately, I've been all about making components scalable and maintaining performance, using tools like Redux and React Query, and typing things out with TypeScript. I also focus on optimizing performance and mentoring junior devs to keep everything up to scratch.

frontend test using Jest and React testing library experience

I've got solid experience with frontend testing using Jest and React Testing Library. I use Jest for unit and integration tests, including snapshot testing to catch unexpected UI changes. With React Testing Library, I test components like a real user would, checking rendering, interactions,

and how they fit with other parts of the app. This keeps the code reliable and closely aligned with how it's used in production.

Experience with REST APIs

I've got plenty of experience working with REST APIs. I've built and consumed APIs, handled authentication, and integrated them into frontend and backend systems. My work includes optimizing API interactions for better performance and ensuring smooth data flow between client and server.

how do you handle user experience and design

I focus on making sure the user experience is smooth and intuitive. I work closely with design teams to implement user-friendly interfaces and ensure that designs are responsive and accessible. I also pay attention to details like performance and interaction feedback to make sure the end product feels polished and easy to use.

"Ability to work in a team, participate in technical and product discussions" experience with it.

I've got a lot of experience working in teams and jumping into technical and product discussions. I actively contribute to brainstorming sessions, share insights on best practices, and collaborate with others to solve problems and refine features. I enjoy exchanging ideas and making sure everyone's on the same page to drive projects forward.

Scrum experience

At Ignite-Reading, I played a key role in implementing Scrum practices. I led sprint planning meetings, facilitated daily stand-ups, and conducted retrospectives to help the team stay on track and continuously improve. I worked on defining user stories and managing the backlog to ensure we focused on high-priority tasks. My involvement in Scrum helped improve our development process, boost team efficiency, and ensure we met project goals effectively.

What is Kanban

Kanban is a workflow management method that helps teams visualize their work, limit work in progress, and maximize efficiency. It uses a board with columns to represent different stages of work (like "To Do," "In Progress," and "Done"). Tasks move through these stages, allowing teams to track progress and identify bottlenecks. It's all about continuous improvement and flexibility in managing tasks.

tell me about redux-based state management

Redux-based state management involves using Redux to handle the state of your application in a predictable way. It uses a single, central store to keep track of the entire application state, and you update this state by dispatching actions and handling them with reducers. This setup makes

it easier to manage complex state interactions, debug issues, and keep state changes predictable across different parts of your app. It's great for maintaining consistency and managing state in large applications.

Tech challenges you faced at your latest role?

One of the most significant challenges I faced in my latest role at Ignite-Reading was refactoring a large codebase from Ruby on Rails to React. This task involved migrating complex business logic and ensuring that the new React components integrated seamlessly with existing systems. It required careful planning and extensive testing to maintain functionality and performance. Also Ruby on Rails was the first time for me. Additionally, I had to manage the transition without disrupting ongoing development and ensure that the new setup met high standards for maintainability and scalability. Overcoming this challenge not only enhanced the application's performance but also provided valuable experience in managing large-scale codebase migrations.

Most tech challenges you faced and how you solved it.

In my previous role at Revolut, I faced a significant technical challenge while developing a high-performance crypto trading platform. The platform needed to provide a seamless and responsive user experience for trading and portfolio management, but we encountered major performance bottlenecks during peak trading times.

The platform's responsiveness was compromised, especially when handling real-time data feeds and executing live transactions. The main issues stemmed from inefficient state management and suboptimal API interactions, which led to slow load times and delayed updates. This was critical since, in the trading environment, even minor delays can lead to significant user dissatisfaction and potential financial losses.

I began by thoroughly analyzing the existing state management setup and API interaction patterns. I identified that the platform was using traditional RESTful APIs for data fetching, which resulted in multiple, redundant calls and unnecessary data processing on the client side.

To resolve this, I proposed and implemented a shift from RESTful APIs to GraphQL. By adopting GraphQL, we were able to fetch only the necessary data, reducing the number of API calls and minimizing the amount of data transferred between the client and server.

I refactored the API interaction logic to batch requests and handle them more efficiently, further reducing the server load and enhancing the responsiveness of the application.

These changes resulted in a 30% increase in platform responsiveness and a significant improvement in user engagement. The optimized performance not only enhanced the user

experience but also reduced the infrastructure costs associated with handling large volumes of data in real time.

How did you manage the state for this challenge?

I first refactored the state structure to ensure components only subscribed to necessary state slices, which minimized unnecessary re-renders. I transitioned to Redux Toolkit for a more efficient state management setup, reducing boilerplate code and aligning with best practices. I also implemented memoization using "useEffect", "useMemo" and etc to prevent redundant calculations and re-renders. To complement these changes, I improved our API interaction by batching requests and updating state more efficiently with the new GraphQL setup. Finally, I used performance monitoring tools to continually assess and refine our state management approach.

Difference between RESTful APIs and graphQL?

First, REST Apis often require multiple endpoints to fetch related pieces of data while graphql allows clients to specify exactly what they need in a single request.

Also REST apis typically require multiple requests to different endpoints to gather related data while with graphql you can aggregate data from multiple sources in a single query.

What did you do after fired?

After my contract ended last month, I took some time off to rest and recharge, particularly to avoid the hot summer. Luckily, this dreadful summer is ending, and this break allowed me to relax and prepare for my next professional opportunity, ensuring I return with renewed energy and focus.

Why did you change companies so many times?

I've changed companies multiple times throughout my career to pursue diverse opportunities for growth and to tackle new challenges. Each transition has been a strategic decision aimed at expanding my skill set, learning from different environments, and contributing to various projects. My goal has always been to advance my expertise and add value wherever I work. For instance, moving from one company to another allowed me to work with different technologies, industries, and teams, which has enriched my experience and prepared me for more complex roles.

How you worked at US companies while you are located at Romania

I've had the opportunity to work with several US-based companies thanks to one of my university friends. A colleague of mine who moved to the US referred me to various companies, which opened up opportunities for me to work on exciting projects and gain valuable experience. Additionally, US-based companies often offer competitive salaries and benefits,

which have been important for my career growth and financial goals. Working remotely for these companies has allowed me to leverage my skills effectively across different time zones while

Git experience

I have extensive experience using Git for version control in all my projects. I'm comfortable with creating and managing branches, handling merges, and resolving conflicts to ensure a smooth development workflow. I regularly use pull requests to review code, provide feedback, and maintain high standards of code quality.

My experience with Git includes tracking changes, managing commits, and using tags to document project milestones. I'm also adept at collaborating with team members through Git, coordinating code contributions, and integrating changes with the main codebase. Additionally, I've worked with various Git tools and platforms like GitHub and GitLab, which helps in managing repositories, code reviews, and continuous integration.

Proven experience as a web developer in similar projects, working closely with designers, product managers, and other engineers;

I have a proven track record as a web developer working on similar projects where collaboration with designers, product managers, and other engineers was crucial. In my recent roles, I've closely worked with designers to translate Figma designs into responsive, pixel-perfect user interfaces. My interactions with product managers have involved understanding project requirements, setting priorities, and delivering features that align with business goals.

Collaborating with other engineers, I've participated in code reviews, shared knowledge, and integrated various components to build cohesive and efficient applications. This experience has honed my ability to work within a team, ensure seamless communication, and contribute to the successful delivery of complex web projects.

the ability to collaborate with product teams and designers to develop the best solution for our users;

I have a strong ability to collaborate effectively with product teams and designers to develop the best solutions for users. My approach involves actively engaging with designers to ensure that their vision is accurately translated into functional and aesthetically pleasing user interfaces. I work closely with product teams to understand user needs, project requirements, and business goals, which helps in prioritizing features and making informed decisions.

Through this collaboration, I aim to find the best balance between user experience, technical feasibility, and business objectives, ensuring that the final product meets the highest standards and provides value to users. This teamwork-oriented mindset has consistently contributed to the successful development and delivery of high-quality web applications.

An understanding of web standards, principles of progressive enhancement, and web accessibility;

I have a solid understanding of web standards, principles of progressive enhancement, and web accessibility. I prioritize following web standards to ensure that my code is compliant with best practices and performs well across different browsers and devices.

With progressive enhancement, I focus on building websites that function and are usable on a basic level for all users, while providing enhanced features and experiences for those with more advanced capabilities. This approach ensures broad compatibility and a good user experience for everyone.

Regarding web accessibility, I make it a priority to incorporate accessibility features into my projects. This includes adhering to accessibility guidelines, such as providing proper semantic HTML, ensuring keyboard navigability, and using ARIA roles and attributes where necessary. This commitment helps ensure that the applications I develop are inclusive and usable by people with various disabilities.

Next.js Experience

I have experience with Next.js, which I've used to build high-performance, server-rendered React applications. In my previous roles, I leveraged Next.js for its server-side rendering capabilities, which improved SEO and initial page load times.

I've worked with Next.js to create dynamic, server-rendered pages and implement static site generation where appropriate. This involved configuring and optimizing Next.js settings, handling API routes, and integrating with various backend services. Additionally, I've utilized Next.js's built-in features for routing, data fetching, and optimizing performance.

This experience has given me a strong understanding of how to use Next.js to build scalable and efficient web applications that deliver a seamless user experience.

React vs next.js

React and Next.js are both powerful tools for building web applications, but they serve different purposes and offer distinct benefits.

React is a JavaScript library for building user interfaces. It focuses on creating reusable components and managing the state of your application. React is excellent for building

single-page applications (SPAs) with rich, interactive user experiences. It provides the flexibility to structure your application as you see fit, but you might need to handle routing, server-side rendering, and other aspects separately.

Next.js, on the other hand, is a framework built on top of React that provides additional features and optimizations. It includes server-side rendering (SSR) and static site generation (SSG) out of the box, which can enhance performance and SEO. Next.js also comes with built-in routing, which simplifies the development process. It offers automatic code splitting, optimized performance, and support for API routes, making it a more opinionated and integrated solution compared to React alone.

In summary, while React is a flexible library for building UIs, Next.js adds server-side capabilities and a set of conventions that can streamline development, improve performance, and enhance SEO for your applications.

what do you think about startup environement?

I find the startup environment to be both exciting and rewarding. The fast-paced, dynamic nature of startups allows me to tackle a variety of challenges and contribute to projects in meaningful ways. I enjoy the opportunity to work closely with a small, dedicated team and have a direct impact on the company's growth and success.

Startups often provide a collaborative atmosphere where innovation is encouraged, and I appreciate the chance to take on diverse responsibilities and learn new skills. While the lack of structure can be challenging, I thrive in environments where I can adapt quickly and help shape the direction of projects. Overall, the energy and potential of a startup are aspects I find very motivating and fulfilling.

AWS experience

I've got solid experience with AWS. I've used AWS Cognito for user authentication, S3 for scalable storage, and Lambda for serverless functions. I'm comfortable setting up and managing these services to ensure applications are secure, scalable, and cost-effective. I've also used various other AWS tools to optimize performance and manage infrastructure, making sure everything runs smoothly and efficiently.

=====

What my previous employers will say about me?

If you contact Ignite-Reading, they'd probably say I'm a reliable and proactive team player who made a positive impact on code quality and efficiency. They'd mention that I helped improve our processes, mentored junior developers, and was always eager to find better ways to do things.

If you contact Ignite-Reading, they'd likely describe me as a proactive and skilled Software Engineer who significantly improved code quality and published private github package to enhance the steam lined development across multiple projects. They would mention my leadership in code reviews, my ability to refactor and enhance components, and my role in adopting modern front-end technologies. They'd also highlight my commitment to mentoring junior developers and fostering a collaborative environment.

how do you update your tech knowledges?

I stay updated with technology through a various of methods: I regularly read tech blogs, follow industry news, and participate in webinars and online courses. I also experiment with new tools and frameworks in personal projects and contribute to open-source projects. Additionally, I engage with developer communities and attend conferences to keep up with the latest trends and practices.

GCP experience

My experience with GCP is limited compared to AWS, but I'm familiar with its core services and concepts. I've managed cloud applications and infrastructure on AWS, which shares similarities with GCP, so I'm confident I can quickly adapt to GCP's environment and leverage its tools effectively.

Build interactive and user-friendly frontend interfaces using Streamlit.

I haven't used Streamlit before, but my extensive experience with React and other frontend technologies will help me get up to speed quickly. I'm confident in my ability to build interactive and user-friendly interfaces and am eager to learn and apply Streamlit's features to create effective solutions.

Collaborate with cross-functional teams to define, design, and ship new features.

I've worked closely with cross-functional teams throughout my career, including at Ignite-Reading and Revolut. I'm skilled in defining project requirements, designing solutions, and ensuring smooth feature delivery. I value clear communication and collaboration to ensure all team members are aligned and that the end product meets everyone's expectations.

Ensure the performance, quality, and responsiveness of applications.

Ensuring performance, quality, and responsiveness has been a key focus in my roles. I implement best practices for code quality, conduct thorough testing, and optimize both frontend and backend components. At Revolut, I improved platform responsiveness by optimizing state management and API interactions, which led to a 30% increase in user engagement.

How do you maintain code quality and organization by implementing best practices?

Maintaining code quality and organization is crucial to me. I follow best practices such as writing clean, modular code, adhering to coding standards, and implementing comprehensive testing strategies. I conduct regular code reviews, use tools for static analysis, and ensure consistent documentation. These practices help keep the codebase maintainable and reduce technical debt.

Your React experience

I have extensive experience with React, starting from my early roles where I built and enhanced user interfaces to my recent positions where I led the development of complex, interactive applications. I've worked with React across different projects, including creating responsive, scalable front-end solutions, integrating with backend services, and optimizing performance. My expertise also includes using Next.js for server-side rendering and Three.js for dynamic 3D visualizations, making me well-versed in building modern, user-centric web applications.

Azure PaaS Infrastructure experience.

I currently do not have commercial experience with Azure PaaS. However, I am familiar with cloud concepts and am confident in my ability to quickly learn and adapt to new technologies. I am eager to expand my skills in Azure PaaS and leverage my strong background in related areas to contribute effectively.

Experience with DDD or Event sourced architecture

I don't have direct experience with Domain-Driven Design (DDD) or Event Sourcing. However, I am familiar with their principles and understand their importance in building scalable and maintainable systems. I am enthusiastic about learning more and applying these concepts in practical scenarios.

Tell me about DDD or Event Sourced architecture

Domain-Driven Design (DDD) is an approach to software development that focuses on modeling complex business domains using domain-specific language and structures. It emphasizes collaboration between domain experts and developers to create a shared understanding of the domain and design solutions that closely align with business needs. Key concepts include bounded contexts, aggregates, and entities.

Event Sourcing is a pattern where state changes in an application are stored as a sequence of events, rather than just the current state. This allows for reconstructing the state by replaying events and provides a comprehensive audit trail. It supports building systems with complex business logic and helps in maintaining consistency across distributed systems.

Tell me about DevSecOps

DevSecOps is an approach that integrates security practices into the DevOps process. It emphasizes the importance of incorporating security at every stage of the development lifecycle, from planning and development to deployment and operations. By embedding security into the continuous integration and continuous deployment (CI/CD) pipeline, DevSecOps aims to identify and address vulnerabilities early, automate security checks, and ensure that security is a shared responsibility across the development team. This proactive approach helps in creating more secure and resilient software, reducing the risk of security breaches and compliance issues.

A strong focus on writing clean, quality code, with a commitment to unit and integration testing

I prioritize writing clean, maintainable code by adhering to coding standards and best practices. This includes following design patterns, ensuring readability, and structuring code to be modular and reusable. I'm also committed to robust testing, both unit and integration, to ensure code reliability and to catch issues early. This approach helps in maintaining high-quality software and supports continuous integration and deployment practices.

how do you think about collaboration with other team members?

I believe effective collaboration is key to a successful project. I make a point to actively communicate with team members, sharing progress and seeking feedback regularly. I value diverse perspectives and work to build strong relationships by being open to others' ideas and offering support where needed. This collaborative approach helps in solving problems more efficiently and ensures that everyone is aligned towards common goals.

how do you resolve if there is language barrier between team members due to english?

To overcome language barriers, I focus on clear and simple communication, using straightforward language and avoiding jargon. I ensure to confirm understanding by summarizing key points and encouraging questions. Additionally, I make use of visual aids, documentation, and tools like translation apps when needed, and I foster an environment where team members feel comfortable seeking clarification and providing feedback.

Proven track record of leading and delivering complex projects from inception to deployment

Yeah, I've led several projects from start to finish. At Ignite-Reading, I refactored a major app from Ruby on Rails to React, improving performance and reducing tech debt. I also built a common components library that streamlined development for the team. At Revolut, I worked on a crypto trading platform, handling everything from frontend architecture to integrating real-time

trading data. I've managed teams, handled deployments, and made sure projects hit their deadlines.

Experience with state management libraries (such as Redux or MobX) and modern front-end build tools

I've worked a lot with Redux, especially in apps that needed complex state management, like the crypto trading platform at Revolut. I've also used React Query for handling data fetching and caching efficiently. As for build tools, I've got experience with Webpack, Vite, and Babel, using them to optimize performance and streamline the build process across different projects.

Ability to architect scalable and maintainable front-end solutions

I've architected front-end solutions that scale well, especially at Revolut where the trading platform had to handle tons of real-time data. I focused on making the codebase maintainable by breaking it down into reusable components and using best practices with TypeScript for type safety. At Ignite-Reading, I also built a common components library that improved scalability and consistency across projects. Always keep an eye on performance and long-term maintainability!

Excellent communication skills with the ability to collaborate effectively within cross-functional teams

I've always been big on communication, especially working in remote teams. At Revolut, I collaborated closely with backend teams to integrate APIs and real-time data for the trading platform. I also worked with designers and product managers to make sure we were all on the same page. Whether it's explaining technical details to non-tech folks or leading team meetings, I'm comfortable keeping everyone in sync.

Ability to work in a fast-paced, agile environment

I'm totally used to fast-paced, agile environments. At Ignite-Reading, we worked in sprints, pushing features quickly and adapting based on feedback. Same thing at Revolut, where we had to move fast to keep up with the market changes. I thrive in Agile, managing tasks efficiently and always staying flexible to handle whatever comes up.

Self-starter who can get up to speed quickly and work independently

I've had to hit the ground running in most of my roles, especially in remote setups. At Ignite-Reading and Revolut, I jumped into complex projects, quickly got up to speed, and started delivering right away. I'm comfortable working independently, setting my own priorities, and driving projects forward without needing much oversight.

Ability to work in a collaborative environment and possess a team oriented attitude

I'm all about teamwork. At Ignite-Reading, I mentored junior devs, did code reviews, and helped create a collaborative culture. I love sharing knowledge and learning from others, whether it's pairing up to fix bugs or brainstorming new features. I believe strong collaboration makes the whole team more productive and keeps the work environment positive.

Do you have experience contributing to open-source projects or active participation in the developer community?

I haven't contributed to open-source projects directly, but I've been involved in other ways, like building and maintaining private libraries and tools. I'm always keen on learning from the developer community and applying best practices from what I see out there. I'm open to exploring open-source contributions in the future as it aligns with my interests in continuous learning and collaboration.

Experience with Jira and Confluence

Yes, I've used Jira for tracking tasks, managing sprints, and overseeing project progress. It's been a key tool in Agile environments for organizing work and collaborating with the team. I've also used Confluence for documenting project details, sharing knowledge, and keeping everyone on the same page. Both tools have been integral in ensuring smooth project management and effective communication.

Experience with AWS technical stack (Cloudfront, API Gateway, Lambda, EC2, RDS)

I've worked with AWS Lambda and RDS quite a bit. At Revolut, I used Lambda for serverless functions, which helped optimize performance and reduce infrastructure costs. I also managed RDS for database needs, ensuring reliable and scalable data storage. While I haven't worked with CloudFront, or EC2 directly, I'm familiar with their roles and can quickly get up to speed with those services if needed.

how do you update the latest-tech trends?

I keep up with the latest tech trends by following tech blogs, joining developer communities, and reading industry newsletters. I also regularly check out updates from major tech conferences and follow influential developers on social media. Staying curious and continually learning helps me keep my skills sharp and stay ahead of the curve.

Your experience implement automated testing and ensure proper documentation of developed features

I've got solid experience with automated testing and documentation. For automated testing, I use tools like Jest and Cypress to ensure code quality and catch issues early. At Ignite-Reading, I set up comprehensive test suites and integrated them into our CI pipeline. As for documentation, I've used tools like Storybook for documenting components and Confluence for detailed project and feature documentation. Both practices help maintain high code quality and keep the team aligned.

What do you think about best practice and code quality?

Best practices and code quality are crucial for building reliable and maintainable software. They ensure that the codebase remains clean, efficient, and easy to understand. This includes following coding standards, writing comprehensive tests, conducting regular code reviews, and keeping the code well-documented. Adhering to these practices helps in catching issues early, improving collaboration, and making the code easier to maintain and scale.

DRY (Don't Repeat Yourself)

- **Definition:** A principle aimed at reducing duplication of code. The idea is to centralize the logic and functionality to avoid redundancy.
- **Example:** Instead of copying and pasting the same function in multiple places, it is good to create a reusable function and call it wherever needed. This makes the code more maintainable and reduces errors.

KISS (Keep It Simple, Stupid)

- **Definition:** Encourages simplicity in design and implementation. The principle emphasizes avoiding complexity and making systems as simple as possible.
- **Example:** If you have two ways to achieve a goal, choose the simpler one if it fulfills the requirements effectively. For instance, using straightforward loops instead of overly complex recursive algorithms when a loop will suffice.

SOLID Principles

- **Definition:** A set of five design principles aimed at making software designs more understandable, flexible, and maintainable. The acronym SOLID stands for:
 1. **S - Single Responsibility Principle (SRP):**
 - **Definition:** A class should have only one reason to change, meaning it should only have one responsibility or job.
 - **Example:** Instead of a class handling both user authentication and data storage, separate these concerns into different classes.
 2. **O - Open/Closed Principle (OCP):**

- **Definition:** Software entities (classes, modules, functions, etc.) should be open for extension but closed for modification. You should be able to extend functionality without changing existing code.
 - **Example:** Use abstract classes or interfaces and inheritance to add new features or behaviors without altering existing code.
3. **L - Liskov Substitution Principle (LSP):**
- **Definition:** Objects of a superclass should be replaceable with objects of a subclass without affecting the correctness of the program.
 - **Example:** If you have a base class `Bird` with a method `fly()`, a subclass `Penguin` should not override `fly()` in a way that breaks the functionality of the base class.
4. **I - Interface Segregation Principle (ISP):**
- **Definition:** Clients should not be forced to depend on interfaces they do not use. Instead, interfaces should be small and specific to a client's needs.
 - **Example:** Instead of one large interface with many methods, create smaller, more specific interfaces that cater to different functionalities.
5. **D - Dependency Inversion Principle (DIP):**
- **Definition:** High-level modules should not depend on low-level modules. Both should depend on abstractions. Additionally, abstractions should not depend on details; details should depend on abstractions.
 - **Example:** Instead of a high-level module directly depending on a low-level module, both should depend on an interface or abstract class.

These principles are designed to help developers create software that is easier to manage, understand, and extend over time.

— Experience —>

Laravel Experience

While most of my recent experience has been focused on JavaScript and React, I've worked with Laravel on several projects in the past. I used Laravel primarily for building robust back-end systems and APIs. One project involved creating a scalable web application where Laravel handled all the server-side logic, database management, and API integrations. I appreciate Laravel's elegant syntax and how it streamlines many common tasks like routing, authentication, and caching. It's definitely a framework I'm comfortable with and can jump back into easily if needed for this role.

React experience

I've got extensive experience with React, having used it in multiple projects over the past several years. I've built everything from single-page applications to complex, interactive web apps using React. I'm well-versed in React's component-based architecture, hooks, and state management using tools like Redux and Context API. One of my key projects involved developing a large-scale web app where React was crucial for creating a dynamic, responsive user interface. React is definitely one of my go-to tools for front-end development.

Ruby on Rails Experience

In my latest role, I had the opportunity to work with Ruby on Rails for the first time. Initially, I was primarily focused on converting Ruby on Rails views to React components, which involved understanding the Ruby backend to ensure seamless integration. As I gained more familiarity with Ruby, I started building backend APIs using Rails. This experience broadened my skills significantly, allowing me to not only contribute to the frontend but also to understand and develop backend functionalities. It was a steep learning curve, but I found it rewarding to build and deploy robust APIs that enhanced our application's performance.

Node/NPM, Webpack and Yarn experience

I've got solid experience working with Node.js and its ecosystem, including NPM, Webpack, and Yarn. Node.js has been a core part of my full-stack development toolkit, especially for building scalable back-end services and RESTful APIs. I've used NPM extensively for managing packages and dependencies in my projects, ensuring everything is up-to-date and working smoothly.

Webpack has been my go-to for bundling and optimizing front-end assets. I've set up custom Webpack configurations to streamline development workflows and improve app performance by managing assets, handling code splitting, and optimizing loading times.

As for Yarn, I appreciate its speed and reliability for dependency management, especially in larger projects where consistency and efficiency are key. I've used Yarn in projects to manage packages, lock dependencies, and streamline the CI/CD pipeline. Overall, I'm very comfortable navigating and utilizing these tools in various development environments.

understanding of JavaScript ES6 features

I have a deep understanding of JavaScript ES6 features, which I've been using extensively throughout my development work. ES6 introduced a lot of great features that have significantly improved the way we write JavaScript. For example, I regularly use arrow functions for cleaner and more concise syntax, and template literals for easy string interpolation.

I'm also a big fan of destructuring, which makes it easier to work with objects and arrays, and I use the spread/rest operator frequently for handling function arguments and combining arrays or objects.

Let and const are my go-to for variable declarations, ensuring better block-scoping and reducing the risk of bugs. I also utilize ES6 classes to create more organized and reusable code, especially when working with complex applications. Lastly, ES6 modules have made it much easier to manage dependencies and keep my code modular and maintainable.

TypeScript Experience

I've been working with TypeScript for a few years now, and it's become a key part of my development process. I really appreciate how TypeScript brings static typing to JavaScript, which helps catch errors early and makes the code more predictable and maintainable.

I've used TypeScript extensively in React projects, taking full advantage of interfaces, generics, and utility types to define component props and state more clearly. It's also been invaluable for working with large codebases, where strong typing helps keep everything organized and understandable.

I've also integrated TypeScript with back-end work, particularly in Node.js environments, where it helps enforce consistency across the application. Overall, TypeScript has been a game-changer in terms of improving code quality and developer experience in my projects.

PostgreSQL Experience

I've worked with PostgreSQL in several projects, mainly for managing relational databases that require robust performance and scalability. I'm comfortable with writing complex SQL queries, designing normalized database schemas, and working with PostgreSQL-specific features like JSONB for handling semi-structured data.

In one of my projects, I used PostgreSQL to manage a large dataset, implementing indexing and optimizing queries to ensure fast data retrieval. I've also set up and managed PostgreSQL databases, handling migrations, backups, and performance tuning. PostgreSQL's reliability and powerful feature set make it my go-to choice for relational databases in many of the applications I've built.

Git Experience

I've been using Git extensively throughout my career for version control. I'm comfortable with the whole Git workflow, including branching, merging, and resolving conflicts. I've managed feature branches, handled pull requests, and conducted code reviews, ensuring that code changes are tracked and integrated smoothly.

I also use Git for coordinating with teams, keeping track of changes, and maintaining a clean and organized codebase. Whether it's for solo projects or collaborative work, Git has been essential for managing code and collaborating effectively.

RESTful API design and implementation

I've got a solid understanding of RESTful API design and implementation. I focus on creating clean and efficient APIs that follow REST principles, such as using proper HTTP methods (GET, POST, PUT, DELETE) and designing resource-based endpoints. I also make sure to handle status codes appropriately to provide clear feedback to clients.

For implementation, I've worked on building APIs using various frameworks and technologies, including Node.js and Laravel. I'm familiar with practices like versioning, rate limiting, and proper authentication to ensure APIs are secure and scalable. I also pay attention to documentation, making sure that API endpoints are well-documented and easy for other developers to use.

Scrum Experience

I've worked in Agile environments using Scrum for several years. I'm familiar with key Scrum practices like daily stand-ups, sprint planning, and retrospectives. I've led teams through sprints, breaking down work into manageable tasks and ensuring that everyone is aligned on goals and priorities.

I've also been involved in managing backlogs, refining user stories, and tracking progress with tools like JIRA or Trello. Scrum has helped me and my teams stay organized, adapt to changes quickly, and deliver high-quality software on a consistent basis.

Most Tech Challenge at your latest role

In my latest role at InnovaTech Ltd., one of the biggest tech challenges was managing the performance of a highly interactive web application that used Next.js and Three.js. The application needed to handle real-time data updates and complex 3D visualizations, which put a lot of strain on the system.

I tackled this by optimizing both the front-end and back-end aspects. On the front-end, I focused on reducing rendering times and improving performance by implementing techniques like virtual DOM updates and efficient scene management in Three.js. On the back-end, I optimized the APIs to handle real-time data more efficiently and used caching to reduce server load.

Another challenge was integrating various third-party services while maintaining a seamless user experience. This involved dealing with different APIs and ensuring that data flowed smoothly between services without causing delays or inconsistencies. I managed this by setting up thorough testing and monitoring to catch and address issues early.

These challenges required a lot of problem-solving and iteration, but they helped improve the application's overall performance and user experience.

AWS experience

I've gained significant experience with AWS during my time at AlterCal, where I used various AWS services to optimize and scale a coaching platform. I utilized AWS Cognito for secure user authentication, AWS S3 for scalable data storage, and AWS Lambda for serverless functions, which reduced infrastructure costs and improved platform performance. My experience also includes deploying and managing applications in AWS, ensuring high availability and reliability across the system.

Amplify Experience

At AlterCall AI, I used AWS Amplify to streamline our development process and manage our cloud infrastructure. Amplify was instrumental in handling authentication, API integrations, and hosting our frontend applications. It simplified the setup of our backend services, which allowed us to focus more on developing features and improving user experience. I found Amplify particularly useful for its seamless integration with React and its ability to rapidly deploy updates.

TypeScript Experience

I have significant experience with TypeScript, particularly in developing responsive user interfaces with React. I've used TypeScript to create reusable UI components, which improved code maintainability and consistency across projects. TypeScript's type safety has been invaluable in reducing bugs and making the codebase more predictable, facilitating smoother collaboration within teams. Overall, it's been a key tool in building scalable and reliable front-end applications.

NodeJS experience

I've worked extensively with Node.js across multiple projects, using it to build scalable and efficient backend services. At AlterCall AI which is based in California, I used Node.js to build robust backend services, creating RESTful APIs that seamlessly integrated with the frontend components.

MySQL experience.

I've worked with MySQL across several projects, particularly at AlterCall AI, where I was responsible for designing and optimizing database schemas. I created and modified tables and schema objects to ensure efficient data storage and retrieval. Additionally, I implemented stored procedures and triggers to automate business logic and streamline repetitive tasks, which

improved overall application performance. My focus has always been on ensuring the database is optimized for both speed and reliability, supporting the scalability of the applications I developed.

TypeScript and NestJS experience

I've used TypeScript extensively at Revolut, where I developed a high-performance crypto trading platform using Next.js. TypeScript's strong typing helped me catch errors early, leading to more reliable and maintainable code. While I haven't used NestJS in a professional setting, my solid background in TypeScript and Node.js gives me the foundation to quickly adapt to and work effectively with NestJS if needed.

Relational database and RESTful API

Yes, I have a strong knowledge of relational databases and REST APIs. I've worked extensively with relational databases like MySQL and PostgreSQL, where I've designed and optimized schemas, created complex queries, and implemented stored procedures and triggers to manage data efficiently.

I have extensive experience with RESTful APIs, having developed, integrated, and maintained them across multiple projects. For example, during my time at AlterCall AI, I built RESTful APIs to facilitate seamless communication between front-end components built with React and back-end services powered by Node.js. These APIs were crucial for managing user data, handling authentication, and integrating third-party services.

In addition to developing APIs, I ensured they were well-documented, secure, and optimized for performance. I also implemented robust error handling and logging mechanisms, which significantly improved the reliability and maintainability of the applications. My work with RESTful APIs has consistently focused on providing efficient, scalable, and secure solutions to meet project requirements.

Ruby on Rails experience.

I have worked with Ruby on Rails primarily in the context of an MVP model where I created some endpoints to interact with a React frontend. However, my main focus wasn't on building the Ruby on Rails backend, but rather on converting Ruby views to React components. I have about 1 year of experience with Ruby on Rails, so while I'm familiar with it, my expertise lies more in frontend development with React.

Collaborating with designers to review and implement UI/UX designs

I have experience collaborating closely with designers to review and implement UI/UX designs. For example, at Revolute, I worked directly with the design team to ensure that the React components I developed matched their specifications and delivered the intended user experience. This involved participating in design reviews, providing feedback on feasibility, and

integrating design assets into the application. I focused on translating design mockups into responsive, functional components while maintaining a high level of visual consistency and usability. This collaboration helped ensure that the final product was both visually appealing and user-friendly.

Collaborative skills with an emphasis on product quality.

Throughout my career, I've placed a strong emphasis on collaboration and product quality. At Ignite-Reading, I worked closely with other team members to ensure our React components aligned with UI/UX specifications and maintained high visual and functional standards. This involved regular design reviews and iterative feedback to refine and enhance the product.

I also actively participated in code reviews and discussions to uphold best practices and high-quality standards. This collaborative approach helped us deliver robust, scalable features and maintain a strong focus on product excellence. By fostering open communication and teamwork, I contributed to creating well-rounded, high-quality products that met both user expectations and technical requirements.

Experience in a distributed work environment.

I have substantial experience working in distributed work environments. In my recent roles at Ignite-Reading, I collaborated with teams spread across different time zones and locations. This involved coordinating with team members remotely, managing tasks through project management tools, and ensuring clear and effective communication. I'm comfortable working asynchronously and am skilled at using collaboration tools like Slack, Zoom, and GitHub to stay connected and maintain productivity. This experience has helped me develop strong organizational skills and the ability to work effectively with diverse, geographically dispersed teams.

Startup companies work experience before?

Yes, I have experience working with startup companies, specifically at Ignite-Reading and AlterCall AI. At both companies, I was involved in a variety of roles and responsibilities due to the fast-paced and dynamic nature of startups. This included developing core features, managing deployments, and adapting to rapidly changing requirements. Working in these environments taught me to be agile and resourceful, collaborate closely with small teams, and handle diverse challenges effectively. These experiences have strengthened my problem-solving skills and ability to thrive in innovative and high-growth settings.

Your think about the startup environment

I find the startup environment both exhilarating and challenging. It offers a unique blend of fast-paced innovation and the opportunity to make a significant impact on the company's growth and direction. Working in startups often means wearing multiple hats, adapting quickly to changes, and tackling a variety of tasks, which fosters a strong sense of ownership and

flexibility. While the pace can be intense, it provides valuable experiences and learning opportunities, helping me to develop a diverse skill set and thrive in dynamic settings.

Professional achievements?

In my latest project at Ignite-Reading, a major achievement was creating and publishing a private GitHub package for a common components library. This library included reusable UI components and utilities that were shared across multiple projects. By centralizing these components, I not only ensured consistency in design and functionality but also streamlined the development process. This initiative reduced redundant work, improved code quality, and increased our team's efficiency. Additionally, it facilitated better collaboration and faster iteration, as team members could easily access and utilize standardized components in their work.

— Challenges—>

Challenges in collaboration and how did you handle this?

One significant challenge I faced in collaboration was overcoming language barriers with non-native speakers. To address this, I focused on using clear, simple language and avoided jargon to ensure mutual understanding. I also utilized visual aids and confirmed key points through follow-up messages to prevent miscommunication. These strategies helped improve clarity and foster more effective collaboration despite language differences.

Tech challenges you faced at your latest role?

One of the most significant challenges I faced in my latest role at Ignite-Reading was refactoring a large codebase from Ruby on Rails to React. This task involved migrating complex business logic and ensuring that the new React components integrated seamlessly with existing systems. It required careful planning and extensive testing to maintain functionality and performance. Also Ruby on Rails was the first time for me. Additionally, I had to manage the transition without disrupting ongoing development and ensure that the new setup met high standards for maintainability and scalability. Overcoming this challenge not only enhanced the application's performance but also provided valuable experience in managing large-scale codebase migrations.

Most tech challenges you faced and how you solved it.

In my previous role at Revolut, I faced a significant technical challenge while developing a high-performance crypto trading platform. The platform needed to provide a seamless and responsive user experience for trading and portfolio management, but we encountered major performance bottlenecks during peak trading times.

The platform's responsiveness was compromised, especially when handling real-time data feeds and executing live transactions. The main issues stemmed from inefficient state management and suboptimal API interactions, which led to slow load times and delayed updates. This was critical since, in the trading environment, even minor delays can lead to significant user dissatisfaction and potential financial losses.

I began by thoroughly analyzing the existing state management setup and API interaction patterns. I identified that the platform was using traditional RESTful APIs for data fetching, which resulted in multiple, redundant calls and unnecessary data processing on the client side.

To resolve this, I proposed and implemented a shift from RESTful APIs to GraphQL. By adopting GraphQL, we were able to fetch only the necessary data, reducing the number of API calls and minimizing the amount of data transferred between the client and server.

I refactored the API interaction logic to batch requests and handle them more efficiently, further reducing the server load and enhancing the responsiveness of the application.

These changes resulted in a 30% increase in platform responsiveness and a significant improvement in user engagement. The optimized performance not only enhanced the user experience but also reduced the infrastructure costs associated with handling large volumes of data in real time.

How did you manage the state for this challenge?

I first refactored the state structure to ensure components only subscribed to necessary state slices, which minimized unnecessary re-renders. I transitioned to Redux Toolkit for a more efficient state management setup, reducing boilerplate code and aligning with best practices. I also implemented memoization using "useEffect", "useMemo" and etc to prevent redundant calculations and re-renders. To complement these changes, I improved our API interaction by batching requests and updating state more efficiently with the new GraphQL setup. Finally, I used performance monitoring tools to continually assess and refine our state management approach.

Difference between RESTful APIs and graphql?

First, REST Apis often require multiple endpoints to fetch related pieces of data while graphql allows clients to specify exactly what they need in a single request.

Also REST apis typically require multiple requests to different endpoints to gather related data while with graphql you can aggregate data from multiple sources in a single query.

— Tech —>

What is important in testing?

In testing, it's crucial to ensure comprehensive coverage and reliability. Key aspects include writing clear and effective test cases that cover various scenarios, both typical and edge cases, to catch potential issues early. Automated testing tools, like Cypress, are essential for maintaining efficiency and consistency across tests. Additionally, integrating testing into the CI/CD pipeline helps catch regressions and ensures that new features do not disrupt existing functionality. Ultimately, effective testing helps deliver a stable and high-quality product.

How JS works?

Yes, I have a deep understanding of how JavaScript works. I'm familiar with its core concepts, including asynchronous programming with callbacks, promises, and `async/await`. I understand JavaScript's event-driven architecture and how it uses the event loop to handle concurrency. My experience includes working with both client-side and server-side JavaScript, which has given me insights into how JavaScript engines like V8 optimize code execution and manage memory. This knowledge helps me write efficient, high-performance code and debug complex issues effectively.

React

1. Difference between `useEffect` and `useLayoutEffect`?

- `useEffect` runs asynchronously after the DOM has been painted. It's used for operations like data fetching, subscriptions, or manually changing the DOM.
- `useLayoutEffect` runs synchronously after all DOM mutations. It's used for reading layout from the DOM and synchronously re-rendering.

2. How to manage state in React?

- **Local State:** Using the `useState` hook for simple, component-specific state.
- **Context API:** For shared state across components.
- **Redux:** For complex state management across many components.
- **React Query:** For managing server state and caching.

3. What are React Hooks?

- Hooks are functions that let you use state and other React features without writing a class. Examples include `useState`, `useEffect`, `useContext`, `useReducer`, etc.

4. How does React's reconciliation algorithm work?

- React uses a virtual DOM to optimize rendering. When a component's state changes, React updates the virtual DOM first, then compares it with the previous virtual DOM to determine the minimal set of changes required to update the actual DOM.

5. What is JSX?

- JSX stands for JavaScript XML. It allows you to write HTML-like syntax in JavaScript, which is then transformed into React elements.

6. How to handle events in React?

- Events in React are handled using camelCase syntax. You pass an event handler function as a prop to a component. For example: `<button onClick={handleClick}>Click Me</button>`.

7. Purpose of keys in React lists?

- Keys help React identify which items have changed, are added, or are removed. They should be unique among siblings to help React optimize the re-rendering process.

8. What are controlled and uncontrolled components?

- **Controlled Components:** Form elements whose value is controlled by React state.
- **Uncontrolled Components:** Form elements that maintain their own state internally.

9. What is `useReducer` hook and when would you use it?

- `useReducer` is a hook used for managing more complex state logic that involves multiple sub-values or when the next state depends on the previous one. It's an alternative to `useState`.

10. What is a Higher-Order Component (HOC)?

- A Higher-Order Component is a function that takes a component and returns a new component with additional props. It's used for reusing component logic.

Node.js

1. What is the Event Loop in Node.js?

- The Event Loop is a mechanism that allows Node.js to perform non-blocking I/O operations. It handles asynchronous operations by executing callbacks and processing events.

2. What are Streams in Node.js?

- Streams are objects that enable you to read data from a source or write data to a destination in a continuous fashion. Types include Readable, Writable, Duplex, and Transform streams.

3. How to handle asynchronous operations in Node.js?

- Asynchronous operations can be managed using callbacks, promises, or `async/await` syntax. Promises and `async/await` are preferred for cleaner and more manageable asynchronous code.

4. Difference between `process.nextTick()` and `setImmediate()`?
 - `process.nextTick()` schedules a callback to be invoked on the next iteration of the event loop, before any I/O operations. `setImmediate()` schedules a callback to be executed in the next iteration of the event loop, after I/O events.
5. How Node.js handle file operations?
 - Node.js provides the `fs` module for file operations. You can perform synchronous and asynchronous operations such as reading, writing, and deleting files using `fs.readFile`, `fs.writeFile`, and other methods.
6. What are middleware functions in Express.js?
 - Middleware functions are functions that have access to the request, response, and the next function in the application's request-response cycle. They can modify the request, response, or terminate the request-response cycle.
7. What is a callback function in Node.js?
 - A callback function is a function passed as an argument to another function, which is then invoked after the completion of an asynchronous operation.
8. Purpose of the `package.json` file?
 - The `package.json` file is used to manage project metadata, dependencies, scripts, and version information. It is essential for defining and managing project dependencies and configurations.
9. Difference between `require` and `import`?
 - `require` is a CommonJS module syntax used in Node.js for importing modules. `import` is an ES6 module syntax that is used for importing modules in modern JavaScript. Node.js supports both, but ES6 modules require specific flags or configurations.
10. How to handle errors in Node.js?
 - Errors can be handled using try-catch blocks, error-first callback functions, or Promises with `.catch()`. For asynchronous code, using `async/await` with try-catch is a common approach for error handling.

DRY (Don't Repeat Yourself)

- **Definition:** A principle aimed at reducing duplication of code. The idea is to centralize the logic and functionality to avoid redundancy.
- **Example:** Instead of copying and pasting the same function in multiple places, it is good to create a reusable function and call it wherever needed. This makes the code more maintainable and reduces errors.

KISS (Keep It Simple, Stupid)

- **Definition:** Encourages simplicity in design and implementation. The principle emphasizes avoiding complexity and making systems as simple as possible.
- **Example:** If you have two ways to achieve a goal, choose the simpler one if it fulfills the requirements effectively. For instance, using straightforward loops instead of overly complex recursive algorithms when a loop will suffice.

SOLID Principles

- **Definition:** A set of five design principles aimed at making software designs more understandable, flexible, and maintainable. The acronym SOLID stands for:
6. **S - Single Responsibility Principle (SRP):**
 - **Definition:** A class should have only one reason to change, meaning it should only have one responsibility or job.
 - **Example:** Instead of a class handling both user authentication and data storage, separate these concerns into different classes.
 7. **O - Open/Closed Principle (OCP):**
 - **Definition:** Software entities (classes, modules, functions, etc.) should be open for extension but closed for modification. You should be able to extend functionality without changing existing code.
 - **Example:** Use abstract classes or interfaces and inheritance to add new features or behaviors without altering existing code.
 8. **L - Liskov Substitution Principle (LSP):**
 - **Definition:** Objects of a superclass should be replaceable with objects of a subclass without affecting the correctness of the program.
 - **Example:** If you have a base class `Bird` with a method `fly()`, a subclass `Penguin` should not override `fly()` in a way that breaks the functionality of the base class.
 9. **I - Interface Segregation Principle (ISP):**
 - **Definition:** Clients should not be forced to depend on interfaces they do not use. Instead, interfaces should be small and specific to a client's needs.
 - **Example:** Instead of one large interface with many methods, create smaller, more specific interfaces that cater to different functionalities.
 10. **D - Dependency Inversion Principle (DIP):**
 - **Definition:** High-level modules should not depend on low-level modules. Both should depend on abstractions. Additionally, abstractions should not depend on details; details should depend on abstractions.
 - **Example:** Instead of a high-level module directly depending on a low-level module, both should depend on an interface or abstract class.

These principles are designed to help developers create software that is easier to manage, understand, and extend over time.

When I entering url to the chrome and hit enter, what happen next?

When you enter a URL in Chrome and hit Enter, several steps occur to load the webpage:

1. **URL Parsing:**
 - Chrome parses the URL to identify the protocol (e.g., HTTP, HTTPS), domain (e.g., `example.com`), and path (e.g., `/page`).
2. **DNS Resolution:**
 - The browser checks if the domain name (`example.com`) has a cached IP address. If not, it sends a DNS query to a DNS server to resolve the domain name to an IP address.
3. **Establishing a Connection:**
 - **TCP Connection:** Chrome establishes a TCP connection to the server using the IP address.
 - **TLS Handshake** (if HTTPS): Chrome performs a TLS handshake to establish a secure connection, involving certificate exchange and key agreement.
4. **Sending an HTTP Request:**
 - Chrome sends an HTTP request to the server, asking for the content of the URL. This request includes headers with metadata like browser type, accepted content types, etc.
5. **Server Processing:**
 - The server receives the request, processes it, and sends back an HTTP response with the requested data (HTML, CSS, JavaScript, images, etc.).
6. **Receiving and Processing the Response:**
 - **HTTP Response:** Chrome receives the HTTP response from the server, which includes status codes (e.g., 200 OK, 404 Not Found) and the requested content.
 - **Parsing HTML:** Chrome parses the HTML document, builds the DOM (Document Object Model) tree, and processes any included CSS and JavaScript.
7. **Rendering the Page:**
 - **CSS Processing:** Chrome applies CSS styles to the HTML elements to render the page.
 - **JavaScript Execution:** JavaScript is executed, which may modify the DOM or perform additional actions.
 - **Layout and Painting:** Chrome calculates the layout of elements (where they go and their size) and paints the pixels on the screen.
8. **Loading Additional Resources:**
 - **Asynchronous Requests:** If the page includes additional resources (like images, scripts, or fonts), Chrome makes additional HTTP requests to load these resources.
 - **Render Updates:** The page may be re-rendered or updated as additional resources are loaded or JavaScript modifies the DOM.
9. **Displaying the Page:**
 - Finally, Chrome displays the fully rendered page to the user.

— Other —>

US hours especially starting at 2-3 pm Ukraine?

Yes, I am comfortable working US hours, starting from 2-3 PM Ukraine time. My most recent role at Ignite-Reading, which is based in California, required me to work in the US time zone for about a year until last month. This experience has prepared me well to manage my time effectively and ensure smooth communication and collaboration with teams and clients across different time zones.

What did you do after fired?

After my contract ended last month, I took some time off to rest and recharge, particularly to avoid the hot summer. Luckily, this dreadful summer is ending, and this break allowed me to relax and prepare for my next professional opportunity, ensuring I return with renewed energy and focus.

Why did you change companies so many times?

I've changed companies multiple times throughout my career to pursue diverse opportunities for growth and to tackle new challenges. Each transition has been a strategic decision aimed at expanding my skill set, learning from different environments, and contributing to various projects. My goal has always been to advance my expertise and add value wherever I work. For instance, moving from one company to another allowed me to work with different technologies, industries, and teams, which has enriched my experience and prepared me for more complex roles.

How you worked at US companies while you are located at Romania

I've had the opportunity to work with several US-based companies thanks to one of my university friends. A colleague of mine who moved to the US referred me to various companies, which opened up opportunities for me to work on exciting projects and gain valuable experience. Additionally, US-based companies often offer competitive salaries and benefits, which have been important for my career growth and financial goals. Working remotely for these companies has allowed me to leverage my skills effectively across different time zones while maintaining a work-life balance from here.

Ur think about the work-life balance?

I believe work-life balance is crucial for maintaining long-term productivity and personal well-being. It's important to have time for both professional responsibilities and personal

interests to avoid burnout and stay motivated. I strive to manage my work schedule efficiently so that I can deliver high-quality results while also enjoying time with family and pursuing hobbies. A healthy work-life balance helps me stay focused, energized, and more effective in my role.

How did u manage work-life balance?

I manage work-life balance by setting clear boundaries between work and personal time. I prioritize my tasks and focus on the most important ones during work hours to ensure productivity. I also make sure to take breaks and disconnect from work after hours to recharge. Effective time management and maintaining a routine help me stay organized and balanced. Additionally, I engage in activities outside of work, such as exercise or hobbies, to relax and maintain a healthy mindset.

Have u heard about us before?

Yes, I have checked your company website before submitting my application. I was impressed by your commitment to high-quality development and the range of interesting projects you handle. It reinforced my interest in the role and motivated me to apply.

What r u looking for in next role?

In my next role, I'm looking for opportunities where I can leverage my frontend development skills to contribute to innovative and impactful projects. I'm particularly interested in working on challenging problems that require scalable and efficient solutions. I want to be part of a collaborative team where continuous learning and knowledge sharing are encouraged. Additionally, I'm looking for a role where I can take on more leadership responsibilities and continue to grow both technically and professionally.

Why u applied to our company?

Because you offer an exciting opportunity to work on diverse and innovative projects with clients from various industries. The role aligns with my expertise in frontend development, particularly with technologies like React, and TypeScript, which are central to your requirements. I'm particularly drawn to the company's commitment to professional growth and continuous learning, as well as the collaborative environment you foster. The flexibility and emphasis on work-life balance, alongside the chance to work with a talented team on cutting-edge solutions, make this an ideal place for me to contribute and grow as a developer.

Your Strength and weakness

Greatest Strength: My greatest strength is my ability to bridge technical and team communication gaps. I excel at translating complex technical concepts into understandable terms for non-technical stakeholders, which helps align project goals and fosters effective collaboration. This skill has consistently improved team efficiency and project outcomes.

Weakness: My main weakness is a tendency to overcommit to tasks due to my enthusiasm for new challenges. While this eagerness often drives innovation, it can sometimes lead to stretching myself too thin. I'm actively working on better prioritization and time management to ensure that I maintain a balanced workload without compromising quality.

1. What is your experience with test libraries?

I have extensive experience with test libraries like Jest and React Testing Library. At Ignite-Reading, I designed comprehensive test suites to validate React components, which improved their reliability and facilitated better team collaboration. Additionally, at Revolut, I used Jest for unit testing and Cypress for end-to-end testing, ensuring high performance and security for a high-traffic crypto trading platform.

2. What is your experience with React and ES6/TypeScript?

I have robust experience with React, ES6, and TypeScript. At Ignite-Reading, I refactored key components from Ruby on Rails to React, enhancing performance and maintainability. I also developed a common components library using TypeScript to ensure type safety and streamline development. My work with ES6 features, such as arrow functions and destructuring, has enabled me to write clean, efficient code.

3. What is your proficiency with Git source control?

I am highly proficient with Git, using it extensively for version control. At Ignite-Reading, I managed branching strategies, code reviews, and pull requests, ensuring smooth collaboration and integration. My expertise with Git has been crucial for maintaining a clean codebase and facilitating efficient team workflows.

4. What is your experience with Tailwind CSS?

I have solid experience with Tailwind CSS, which I used to create responsive and efficient user interfaces. At AlterCall AI, I leveraged Tailwind to prototype and style components quickly, leading to faster development cycles and a consistent design across the application.

5. How would you describe your communication skills in written, verbal, and code contexts?

I excel in communication across written, verbal, and code-based mediums. I document technical concepts clearly, ensure that stakeholders understand project details, and lead team discussions effectively. In code, I use clear naming conventions and comments to make my work easily understandable and maintainable.

6. What is your experience designing solutions from the ground up?

I have extensive experience designing solutions from scratch. For instance, at Revolut, I engineered a high-performance crypto trading platform, defining the architecture, tech stack, and key features. Similarly, at Ignite-Reading, I led the migration of components to React, focusing on performance and maintainability.

7. What is your experience working with teams to define and use APIs?

I have significant experience working with teams to define and integrate APIs, often using stubbed APIs for early development. At Revolut, I collaborated on defining GraphQL APIs and used stubbed versions to allow frontend development to proceed smoothly while the backend was still being built.

8. How do you approach code reviews of your team members?

I actively participate in code reviews, focusing on maintaining high standards and promoting best practices. At Ignite-Reading, I led code reviews, providing constructive feedback and ensuring adherence to coding standards, which improved code quality and served as a learning opportunity for junior developers.

9. How do you collaborate with team members on coding tasks?

I regularly pair with team members to collaborate on coding tasks and solve complex problems. At Ignite-Reading, I engaged in pair programming to tackle challenging features, which enhanced problem-solving and facilitated knowledge sharing among the team.

10. How do you handle troubleshooting?

I approach troubleshooting by systematically analyzing issues, reproducing problems, and isolating root causes. At Revolut, I addressed performance bottlenecks and critical bugs by profiling code and implementing optimizations, ensuring application stability and performance.

11. How do you take responsibility for stories from inception to delivery?

I take full responsibility for stories from inception to delivery, managing each feature's development lifecycle. At Ignite-Reading, I handled everything from initial planning and design to implementation and deployment, ensuring that each story aligned with business goals and met project requirements.

12. What is your experience working with product managers and stakeholders?

I have substantial experience collaborating with product managers and stakeholders to align on project goals. At Revolut, I worked with product managers to define and prioritize features, ensuring development met both user needs and business objectives through regular communication and updates.

13. How do you help your teammates improve their skills and knowledge?

I am committed to helping my teammates enhance their skills and knowledge through mentorship and collaborative learning. At Ignite-Reading, I conducted code reviews, organized knowledge-sharing sessions, and provided one-on-one mentoring, contributing to the team's overall technical growth.

Intro Call

1. Tell me about yourself.

"I am a positive, enthusiastic and competent Web Developer who, over the years, has built up a diverse range of skills, qualities and attributes that guarantee I will perform highly in this role. I have extensive experience working both alone and as part of a team on often time-sensitive, challenging web development projects that require outstanding creative and technical capabilities and the ability to ensure all work is optimized across a wide range of platforms. I take my work as a Web Developer seriously and this means I always ensure my skills are kept up to date within this rapidly changing industry. If you hire me as your Web Developer, I assure you I will fit into your team quickly, I will always put the commercial needs of your business at the forefront of everything I do, and the work I carry out will be consistent to a first-class standard."

"Let me introduce myself in 7 words. Responsible, creative, open-minded, friendly and ambitious web developer. This is how I see myself, and hope you will give me a chance to prove my words in the job, or at least in this interview.

I have been developing websites, mostly in MERN stack, for 5 years already. Ranging from small personal projects to contracts worth of tens of thousands of dollars, I had developed or helped to develop dozens of websites. Now, however, I'd like to belong to some bigger team, bigger organization, which will allow me to work even on bigger projects and have a bigger impact. That's why I decided to apply for a job with your company, and hope for a chance in the interview.

Oh, and besides that I am a normal guy who enjoys hiking and spending time with friends, while not working or studying, which takes most of my time of course."

----- (experience)

"I am working as a MERN stack developer for 5 years and developed many projects as an individual and as a lead developer for some companies and clients.

Throughout my career, I've worked on various projects that involved implementing cutting-edge solutions and driving successful outcomes.

For instance, in my previous role, I led a team responsible for developing Pirate Rush, a Solana casino game. This experience allowed me to hone my skills in React and Node more as long as Ethereum and Solana Web3 while delivering tangible results.

If you are looking for experienced MERN stack developer, I believe I would be a great fit for your company.

"

2. How did you hear about this position?

I found this position on a job search website while looking for web development opportunities in the area. The job description and company culture seemed like a great fit for my skills and career goals, so I decided to apply.

3. Why do you want to work for our company as a Web Developer?

"There are lots of web developer jobs out there, but I want one where I am going to be challenged and where the skills and qualities that I have built up are going to be put to good use. I carried out lots of research into your organization prior to submitting my resume/CV, and the work you do, the products and services you offer, and the manner in which you support your employees tell me this is a great place to work. I am seeking long-term employment with the same company where I can really immerse myself into their brand and their online ambitions, and this is genuinely the only place I want to work as a Web Developer."

4. Why should we hire you?

"I believe that my experience in web development, specifically in the MERN stack development, makes me the best match for this position. In my previous job, I was responsible for maintaining and updating the casino game website as a lead developer. This required keeping employee profiles updated and continuously posting information regarding upcoming events. I truly enjoyed what I was doing, which is what drew me to apply for this position. I would live to bring my skills I learned there to this new and more challenging role."

5. What will you bring our company?

As a web developer, I am passionate about creating websites and applications that are not only functional but also beautiful and engaging. I have a solid background in front-end development, using HTML, CSS, JavaScript, React, and MUI to create responsive and interactive web pages. I also have some experience in back-end development, using Node.js, Express and MongoDB to create RESTful APIs and database-driven applications. I am always eager to learn new skills and tools that can help me improve my work and solve problems. I have excellent problem-solving and debugging skills, and I can work independently or as part of a team. I would love to work for your company because I believe that your projects are challenging and rewarding, and I think that I can contribute to your success with my skills and enthusiasm.

6. What do you like and dislike the most about web development?

As a web developer, I love being able to create something from scratch and seeing it come to life on the internet. It's also very satisfying to solve complex problems and improve website performance. On the other hand, I dislike dealing with difficult clients who don't understand the technical aspects of web development and unrealistic deadlines that don't allow for proper testing and debugging.

I really enjoy the creative aspect of web development, the ability to bring designs to life and solve complex problems. I also appreciate the constant learning and the challenge of keeping

up with new technologies and trends. On the other hand, sometimes tight deadlines can be stressful and balancing multiple projects can be challenging. However, I've learned to manage my time effectively and prioritize tasks to ensure that projects are completed on time and to a high standard

7. What is the last project you developed?

I recently completed a project developing a Solana casino game, which was finished about a month ago. The project involved designing and developing a casino-style game that was built on the Solana blockchain. I led a team of developers to create a game that was both engaging and easy to use. We used a range of technologies, including React, TypeScript, and Solana Web3 library, to build the game from scratch. One of the key challenges we faced was ensuring that the game was secure and that users' funds were protected. To address this, we implemented a range of security measures, including multi-factor authentication and encryption. The project also required us to work closely with the client to understand their vision and requirements, and to ensure that we delivered a product that met their expectations. Overall, I am proud of the work we did on this project and believe that it demonstrates my ability to develop innovative and secure solutions using cutting-edge technologies."

8. Describe a time when you failed as a web developer?

During a previous project, I was tasked with developing a new feature for a client's website. I was confident in my abilities and didn't ask enough questions about the requirements. As a result, I misunderstood the client's needs and developed a feature that didn't meet their expectations. When I presented the feature to the client, they were disappointed and it caused delays in the project timeline.

However, I learned from this experience and realized the importance of clear communication and asking questions to fully understand the requirements. I took responsibility for my mistake and worked closely with the client to make the necessary changes to the feature. In the end, the project was completed successfully and the client was satisfied with the final result.

One time, I was working on a project with a tight deadline and I didn't properly test a feature before launching it. As a result, the feature had a bug that caused some users to have trouble minting NFTs on website. I immediately took responsibility for the mistake and worked with my team to fix the issue as quickly as possible. We also implemented a more thorough testing process to prevent similar issues from happening in the future. While it was a challenging experience, it taught me the importance of testing and quality control in web development.

9. In your experience, what are some of the biggest challenges faced when working on the frontend(or backend) of an application?

-----Backend-----

In my most recent project, I successfully developed a Solana casino game using the MERN stack with Web3 integration. Although the website development process was relatively straightforward, I faced significant challenges in ensuring the website's security against potential hacks. Through this project, I gained valuable experience and knowledge in implementing robust security measures to safeguard web applications.

-----Frontend-----

In my latest role, one of the biggest challenges I faced was publishing a private GitHub package and integrating it across multiple projects. The challenge was ensuring that the package was flexible enough to meet the varying needs of different teams while maintaining consistency and high code quality. To resolve this, I focused on creating a well-documented, modular package with clear guidelines for usage. I also collaborated closely with other teams to gather feedback and made iterative improvements based on their needs. This approach not only streamlined the development process but also improved overall team efficiency.

10. Where do you see yourself in 5 years?

As a web developer, in five years I see myself as a more experienced and skilled professional, having worked on a variety of challenging projects and expanded my knowledge of different programming languages and frameworks. I hope to have taken on more leadership roles and mentored junior developers, while also continuing to learn and stay up-to-date with the latest web development trends and technologies. Ultimately, my goal is to become a respected and successful web developer who is known for creating innovative and user-friendly web applications.

11. What is your biggest strengths?

My biggest strength as a web developer is my ability to learn quickly and adapt to new technologies and frameworks. I enjoy staying up-to-date with the latest trends and best practices in web development, and I am always eager to expand my knowledge and skills. Additionally, I am detail-oriented and take pride in writing clean, efficient, and maintainable code. I am also a good problem solver and enjoy working collaboratively with other developers and stakeholders to find creative solutions to complex challenges.

12. What is your biggest weakness?

As a web developer, one of my biggest weaknesses is my tendency to get too focused on the details of a project and lose sight of the bigger picture. While attention to detail is important in web development, it can also be a hindrance if it leads to delays or prevents me from making progress on the project as a whole. To address this weakness, I try to learn to set clear priorities and deadlines for myself, but still this is my biggest weakness.

13. What are your salary expectation in this Web Developer role?

Thank you for asking about my salary expectations. While compensation is important to me, I'm more interested in finding a role that's a good fit for my skills and experience, and where I can make a meaningful contribution to the team.

14. Do you have any other question for us?

Thanks for giving me a chance to ask more about the company.

I would like to know what are the most important qualities you're looking for in a candidate for this role.

Also will the role be the remote, in-office or hybrid?

Can you tell me about the team I'll be working with?

Okay, that's all. No more question at the moment.

15. What is your greatest professional achievement?

My greatest professional achievement is my latest project leading a team of developers to launch a complex crypto casino game. I was responsible for managing the project timeline, coordinating with stakeholders, and ensuring the website's functionality and user experience met the client's requirements. It was challenging project, but the end result was a great success and a testament to my team's hard work and dedication.

16. Tell me about a challenge or conflict you've faced at work and how you dealt with it.

During the development of my Solana casino game, it was unfortunately subjected to cross-site scripting (XSS) attacks resulting in the loss of over 50 SOLs. At the time, I was not familiar with XSS and its potential impact. However, I was able to take measures to secure the site and prevent further hacking attempts with input validation. Moving forward, I understand the importance of staying vigilant about security threats and keeping up with best practices in web development to ensure the safety and security of my website.

17. Why are you leaving your current job?

Firstly, I completed all of my responsibilities and tasks within my previous role. Secondly, I am seeking new challenges and opportunities for growth in a larger-scale project and company. Therefore, I made the decision to leave my previous job and pursue these goals, which brings me to where we are today.

18. Why were you fired?

19. What are you looking for in a new position?

I am actively seeking a new position that provides ample opportunities for professional growth and development. Ideally, I am looking for a collaborative work environment where I can work alongside team members who share my passion for excellence and innovation. Additionally, I am seeking a company culture that aligns with my values and work ethic, where I can thrive both personally and professionally. I am particularly interested in a role that allows me to utilize my skills and experience to make a meaningful impact on the organization, whether through contributing to important projects or driving positive change within the company. In summary, I am excited to explore new opportunities that will allow me to continue to grow and develop as a professional while making a valuable contribution to a dynamic and forward-thinking organization.

20. What type of work environment do you prefer?

"I thrive in a work environment that values collaboration and open communication. As a web developer, I enjoy working on projects with a team and appreciate receiving feedback that helps me improve my work. I also value a company culture that encourages innovation and creativity, where employees are empowered to take ownership of their work and contribute to meaningful projects. Additionally, I prefer a work environment that offers opportunities for professional growth and development. Ultimately, I believe that a positive and supportive work environment is essential for achieving both personal and organizational success."

21. What's your work style?

My work style is detail-oriented, and I strive to deliver high-quality work that meets the needs of the end-users. As a web developer, I am proactive and take ownership of my work, ensuring that I meet project deadlines and exceed expectations. I enjoy collaborating with others and working on projects with a team, where I can share ideas and learn from others. However, I am also comfortable working independently and taking the lead on projects when required. Ultimately, my work style is driven by my passion for creating innovative and user-friendly websites or applications that deliver a great user experience.

22. What makes you unique?

What makes me unique as a web developer is my ability to think creatively and solve complex problems. I have extensive experience working with various programming languages and technologies. I am passionate about staying up-to-date with the latest web development trends and techniques, which allows me to bring fresh ideas and perspectives to my work. Additionally, I am highly detail-oriented and have a strong work ethic, which enables me to deliver high-quality work that meets the needs of the end-users. Whether working independently or as part of a team, I am committed to delivering innovative and user-friendly websites and applications that exceed expectations.

23. What is your last position and why you were fired?

24. Give us an overview of your experience with React.

Sure! I have been working as a React frontend developer for the past 5 years. During this time, I have built several web applications using React, Redux, Hooks and other related technologies. I am also familiar with modern frontend development tools and best practices.

For example, in my last project, PirateRush, a crypto casino game built on Solana network, I built website with React Hooks and Material UI to make frontend responsive.

25 Tell me your experience about developing SPAs on microservices architecture.

During the development of PirateRush game using React, I had the opportunity to design and build Single Page Applications (SPAs) on a microservices architecture. This involved employing a range of sophisticated techniques to create an immersive gaming experience that was both seamless and scalable. By leveraging microservices architecture, I was able to break down the game's functionality into smaller, more manageable components, resulting in a highly efficient and flexible system. As a result, PirateRush is a testament to my ability to develop complex applications utilizing cutting-edge technologies, and I am confident that my expertise in this area would be a valuable asset to your team.

26. What types of project would you like to work on?

I am interested in pursuing a project involving the MERN stack and Blockchain integration. I am confident in my ability to build the project from the ground up, and I am eager to take on the challenge of designing and establishing the entire project.

27. What interests you the most about building a freelancer career?

As a dedicated freelancer, I am committed to leveraging my skills and expertise to overcome a multitude of challenges for prominent companies. By taking on diverse projects and building a unique professional experience, I aim to establish myself as a reliable and innovative solution provider in the industry.

28. When working with a client remotely, what do you consider the key to success?

When working with a client remotely, there are several key factors that I consider crucial to achieving success.

First and foremost, clear and consistent communication is essential. This means establishing a regular means of contact, such as video conferencing or messaging, and ensuring that both parties are on the same page regarding project goals, timelines, and deliverables.

Secondly, setting realistic expectations is important. This involves being transparent about my capabilities and availability, as well as clearly defining project scope and limitations.

Lastly, I prioritize delivering high-quality work that meets or exceeds the client's expectations. This involves being proactive in seeking feedback and making necessary revisions throughout the project lifecycle. By prioritizing clear communication, realistic expectations, and high-quality work, I believe that successful remote collaboration with clients can be achieved.

29 What kind of role are you looking for in a new position?

Thank you for asking. Based on my experience and skills, I am looking for a role that allows me to utilize my strengths in project management and data analysis. I am excited about opportunities where I can work collaboratively with a team to achieve common goals and contribute to the growth of the company. In the long-term, I am looking for a position that can

challenge me and provide opportunities for professional development. I believe that this position aligns well with my career goals and I am eager to learn more about the role and how I can contribute to the team.

----- (noted) -----

30 What projects did you developed after last position?

After the last position, I worked as a freelancer and have developed several web and blockchain applications.

The biggest of them is PirateRush, the crypto casino game built on Solana network. I led a team of developers to create a game that was both engaging and easy to use. We used a range of technologies, including React, TypeScript, and Solana Web3 library, to build the game from scratch. One of the key challenges we faced was ensuring that the game was secure and that users' funds were protected. To address this, we implemented a range of security measures, including multi-factor authentication and encryption. The project also required us to work closely with the client to understand their vision and requirements, and to ensure that we delivered a product that met their expectations. Overall, I am proud of the work we did on this project and believe that it demonstrates my ability to develop innovative and secure solutions using cutting-edge technologies.

I also had the opportunity to work on a project called Blockswap, which is a decentralized exchange platform built on top of the Uniswap v3 interface. I leveraged my skills to customize the platform's design, add new features, and ensure seamless integration with the Uniswap v3 protocol. Blockswap enables users to swap tokens from one blockchain network to another, leveraging the advanced liquidity and trading capabilities of the Uniswap v3 protocol. Throughout the development process, I collaborated with a team of designers and developers to ensure the platform met the highest standards of security, scalability, and user experience. Overall, Blockswap represents a new frontier in decentralized finance and I'm proud to have contributed to its development.

31 What industries have you worked on previous companies?

- "In my previous company, I had the opportunity to work on a variety of website projects. One notable website I built was an e-commerce platform for a retail company. The website had a user-friendly interface, integrated payment gateways, and a robust inventory management system. It also featured personalized product recommendations and a seamless checkout process. Throughout the development process, I collaborated closely with designers, developers, and stakeholders to ensure the website met the client's requirements and delivered an exceptional user experience."

- "In my previous company, I was involved in building a social networking website. This platform aimed to connect individuals with similar interests and foster a sense of community. It included features such as user profiles, a news feed, messaging functionality, and the ability to create

and join groups. The website also had robust privacy settings to ensure user data protection. Throughout the development process, I collaborated with a cross-functional team to design and implement the website, focusing on scalability, performance, and user engagement."

- "In my previous company, I had the chance to work on a corporate website for a technology firm. This website was designed to showcase the company's services, highlight their expertise, and attract potential clients. It featured a modern and responsive design, with a clean layout and intuitive navigation. The website also included a blog section where the company could share industry insights and updates. I worked closely with a team of designers and developers to ensure the website was visually appealing, optimized for search engines, and delivered a seamless user experience across different devices."