

Javorko Cvokic

Full-Stack and Blockchain Engineer

javsfddev323@gmail.com

Education

2012 – 2016 GPA 3.9

University of Novi Sad
Bachelor of Computer Science

Contact

 Belgrade, Serbia

 +1 (949) 607 8876

 in / javorko-cvokic-6b8b49272

 github.com/aidear323

Language

Serbian 

English 

Interests

- Football
- Golf
- Tennis
- Swimming

Summary

As a skilled and versatile frontend developer with 6 years of experience, I have a strong track record of building responsive and user-friendly web applications. I am a detail-oriented problem solver with a passion for clean and efficient code. I am constantly learning and adapting to new technologies and best practices in order to deliver high-quality solutions to complex problems.

Work Experiences

2022.11 - 2023.02

Smart Contract & Frontend Developer at Ragratz



- Developed Onchain-Offchain Combination Project on Terra platform.
- Supporting users with simple actions to get incentive earnings with high APY, and transfer fund from bank account to Blockchain with One-Click.
- Developed Defi Project on BSC platform. Wrote contracts for Dex, Staking, Lottery and Bridge, and worked with Audit company for releasing project to world.
- Developed Launchpad contract and UI and worked with Token Sellers and Marketing Managers.

2022.03 – 2022.08

Full-Stack Engineer at Pawoo



- As a project manager, developed a dashboard that makes project management easier and accurately evaluates the results of each developers.
- Built many websites using React, Redux, NodeJS, MongoDB
- Improved SQL Server performance with stored procedure refactors and new constraints and indexes
- Used GraphQL and REST API and use Mocha, Jest, Karma, Jasmine (mainly Jest and Mocha) improved for ensures that different tests don't influence each other's results.
- Development of Electron App and Used Next.js and Nuxt.js - Web Hosting to AWS / Azure
- Design and Create RESTful Web APIs and Integrate with Front-end

2021.02 – 2021.05

Senior Frontend Developer at FastSwap



- Investigate and implement new design patterns, professional front-end architecture, high-quality performance and security using SPA JavaScript Frameworks – React/Next.js, Vue/Nuxt.js and Angular
- Combine SSR (Server Side Rendering) and SSG (Static Site Generating) and improved performance using Next.js
- Adapting designing pattern to business needs

My Skills

UI/UX Design

- Figma
- Adobe XD
- Adobe Photoshop

Front-end skill set

- React, Next, React Native
- Vue, Nuxt
- Angular

Back-end skill set

- Node, Express, Nest
- Laravel
- Python, Django

Database skill set

- MySQL, SQLite, Postgres
- MongoDB, Firebase

BlockChain skill set

- Web3.js, Ether.js
- Solidity
- Hardhat, Infura, Moralis API

Project Management Tool

- AWS
- Github, Gitlab, Bitbucket
- Jira, Trello

2017.02 – 2019.01

Front-end Developer

at Privi



- Implemented Single Page Application (SPA) architecture and client side MVC using ReactJS
- Used Bootstrap features such as model objects, Tabs, Tooltips, Carousals throughout the application
- Implemented components and wire-frames using cross-browser compatible CSS, TypeScript, jQuery and AJAX, React JS
- Used Redux, React Hook, SSR to refresh only certain section of the page and reduce the load on the web server
- Developed many mobile apps using React Native, Flutter and Firebase

Recent Projects

<https://www.metabountyhunters.io>

Company : UVIK Software **Role :** Front-end developer **Period :** 4 months

• **Complex Problems**

Because of the improvement in the loading speed of pages, especially the load of assets files such as pictures and video files, and the load of data, the correlation between pages did not proceed accurately.

• **How to solve**

- To solve the data loading problem, only the data necessary for the displayed page was taken from the service and stored in Redux when loading the page, and only the data to be modified was sent to the service when it was necessary to update.
- SSR was implemented to improve page loading speed, and headless CMS was used to increase the loading speed of all assets.
- All assets were stored in the Strapi server, and GraphQL was used to bring only the necessary assets and blogs when loading the page.

<https://ticketvoordeel.nl>

Company : ZedPro.Me **Role :** Back-end developer **Period :** 7 months

• **Complex Problems**

It is to realize database optimization for large-capacity database management.

• **How to solve**

- implemented normalizing a database
Normalizing database is a logical outcome when designing a relational database, but some developers still decline this process at times. Optimizing databases reduces the amount of space that a database requires and organizes your data to improve website performance.
- used database indexes
A common issue that slows down your website's performance is database indexes. The primary purpose of creating database indexes is to search for matching data instead of going through all the searches in the database.