

GORAN MRŽLJAK - REACT/NODE DEVELOPER

Zagreb, Croatia

(+385) 91 224 3145 ♦ goran.mrzljak@gmail.com

INTRO

Software Developer with more than 14 years of professional experience. Most of my experience is doing web development. Currently, my primary focus is on React/Node stack.

Open to doing any of the following: full stack development, frontend development, backend development.

Also have experience with onboarding new developers, mentoring and interviewing new candidates.

WORK EXPERIENCE

Freelance software developer/Self employed, Remote

January 2016 - Present

Full Stack Web Developer

A very large project used for managing the business and operations of a market leading mobile fueling company from California (company has several hundred employees and exponential growth). Project involved dozens of developers. As part of the system, there are several web applications built with React and Angular, mobile applications (both iOS and Android), with backend being mostly written in Node (Express):

- Tech used: Angular, NgRx, Typescript, Mocha, Node, Express, Nx, ESLint, Prettier, Mongo.
- My role: I was a senior full-stack developer (with more focus on the backend). In addition to technical work, I was also involved with onboarding and mentoring new employees, and interviewing potential candidates. Our work was on a system that is fully in production, and company depends on it working. We implemented the practice of continuous integration. In addition to adding new features and solving technical issues, I had ownership of disparate parts of the system, I was involved (and sometimes was the primary evangelist) in initiatives related to code quality improvements, such as changes to the way we do unit testing and introduction of monorepo. On occasion I was required to do code changes that would effect the functioning of the entire system, and (in cooperation with other developers and PMs) was able to do that without significant issues. I was also required to do occasional migrations on a production Mongo database, and with careful testing and work process was able to do that without any errors during my time at the company. Most of my technical work was on the backend node application (all the way from the API down to the database), and a smaller part of my job was working on an admin web app written in Angular.

A medium sized web application (100 kloc frontend) in the insurance domain that is built for a UK client. The app facilitates connection of different parties involved in the insurance process, search of insurance schemes etc:

- Tech used: React, TypeScript, Jest, Webpack, Storybook, React Testing Library, Material-UI, Redux/redux-observable/RxJS, Cypress.
- My role: My role: I got onto the project after it was in development for about a year and half. For most of my time on the project, I was the sole frontend developer. My job was refactoring large chunks of the frontend code base, from the API interface to the UI. This included: refactoring of components and entire app sections, fixing application-wide typing issue, complete refactoring of the state management system and async (API calling) infrastructure (using Redux and redux-observable), cleaning-up top level app structure, clean-up and changes to the development/build toolchain (scripts, Webpack), introduction of linting, some tests (all 'utility' code was well covered by tests, but I did only basic testing for component and a handful of Cypress tests - login process

etc). I included Storybook and used it to create a component library (sort of like a wrapper around Material-UI) that matched existing app styles to reduce the existing code duplication. And of course, I did new feature development and fixes when necessary.

A large application for accounting and transactions built for a medium sized Finnish company. The app contains dozens of screens with very complicated logic, forms and data display/editing:

- Tech used: Angular, TypeScript, Jest, RxJS, ag-Grid.
- My role: I worked exclusively on the frontend. A colleague and I did not get on board of this project from the start, but we did get on it from the start of any sizeable development efforts (there was some very sporadic development before, by a single person I believe, and the project was in some kind of a prototype phase). The two of us did most of the work in the beginning, and we were able to make some architectural choices at this time, but the team quickly grew over the next few months to (I think) more than 10 people on the frontend alone. After this, my role was essentially that of a senior developer. I was responsible for some application-wide changes, some subsystems/patterns that were reused throughout the project, and also some larger features and screens, down to simpler maintenance work when required.

Small app for managing graveyards, adding/editing deceased/grave owner/grave data etc. Among other things, it has clickable maps for graveyards, implemented using OpenLayers:

- Tech used: Vue, TypeScript, Jest, OpenLayers, Spring Boot, Kotlin, Liquibase, PostgreSQL, H2, Jasper Reports, Docker, Docker Swarm.
- My role: I implemented almost the entire app. For frontend, I used Vue/TypeScript, with ESLint linting. For backend, I used Spring Boot/Kotlin, Liquibase for database setup and migrations and PostgreSQL for database (H2 for development and automated tests). As an experiment, and since it is more flexible and more familiar to me, I implemented detailed tests for the backend using Jest/TypeScript, along with the scripts to set everything up (build project and run server if necessary). Obviously, this is not super fast, but it tests the entire backend stack and covers a large part of the functionality - works great for my case. There are no automated tests for the frontend. I also used Docker and Docker Swarm on the project, but only in development to automate certain tasks and try out a more automated deployment process, but in the end the production deployment was still being done manually.

A mobile app for buying parking tickets in the city of Zagreb (capital of Croatia). It is for use by the general public:

- Tech used: Java, and a company-internal framework for web development.
- My role: I implemented the "middle" part of the app - the part that connects all the other parts of the app, and is also the top layer of the backend. This part includes the web API (REST), to be used by the mobile frontends (Android and iOS), backend validations, and it directly interfaces with parking service provider APIs. It also communicates with the payment providers (indirectly, through a component built by another person for this project), and to the data access layer, also built by someone else.

Interface to the government's "National identification and authentication system" (NIAS) built for a client (Croatian Post Office). NIAS is a Single Sign-on system. This interface allows existing users of NIAS to log in to the client's system, and allows the client to provide credentials for its users to access dozens of other services that are part of the NIAS system (so it works two ways):

- Tech used: Java, JSP, SOAP.
- My role: y role: I did the entire implementation of this interface.

A component that is used for validation of payments to government, government agencies, local/regional administrations etc. Various parts of the invoice (description, destination account, dates, "model" and "call-to-number" fields, etc.) needed to be validated against several hundred possible rules of several

distinct types. The validation rules needed to be configurable, and for this I used a json file several thousand lines long, to account for all of the needed checks. The component is used as a part of larger system(s). Conceptually, the component works simply by accepting a structured list of invoice data, and outputs a structured list of validation errors and warnings (if any) with all the relevant data for each issue:

- Tech used: Java, JUnit.
- My role: I created the entire component with extensive automated tests. Since the inputs and outputs were so clearly defined, and the validation logic quite complicated, this is probably the best example of an application where automated tests are so easily devised and so invaluable. The first batch of detailed manual tests done by QA revealed a few errors (in a very large number of test scenarios). I was given a very well structured Excel file with results, which I was easily able to convert to a CSV usable for testing. So all of those tested scenarios (the ones that already passed, and the ones that failed) became part of the testing suite. All I needed to do was make the changes so that all of the tests pass, which was very easy at this point. There were no further errors that I know of.

EU Roaming usage and traffic tracking system for the largest Croatian telecom:

- Tech used: Java, Maven, Spring Boot, JPA/Hibernate, Querydsl, Liquibase, Jasper Reports.
- My role: Implemented almost all of the backend for the Angular web application used for administration of this tracking system, from the REST API down to the repository/data access layer. This includes JPA and Querydsl queries as well as some parts of the database, mostly related to the admin application itself (such as account management and account related data).

e-learning web application (MVP phase):

- Tech used: AngularJS, Java, Maven, Spring Boot, JPA/Hibernate, QueryDsl, PostgreSQL, Video.js (JavaScript video player framework), Veeting Rooms (video conferencing platform).
- My role: Besides some initial work by other developers, I implemented the entire application. I also handled deployment, communication with clients and I was actively involved in the development of requirements.

Simple service used for exporting data from database in JSON format, built using C# and WCF. The target database is Microsoft SQL Server.

Some 'emergency' work on an Alfresco based web app for a client very close to production deadline.

Work on backend for a talent acquisition software built using Ruby.

Small project used for talent management (HR/recruitment) built with React.

APIS IT, Zagreb (Croatia) *C# Developer*

November 2013 - May 2016

Large scale (1M+ daily page views) e-Tax web application made for Croatia's Ministry of Finance. This is the application that allows certain categories of Croatia's taxpayers to submit tax forms online:

- Tech used: C#, ASP.NET Web Forms, jQuery, Microsoft Reports (.rdlc), IBM DB2.
- My role: I implemented 7 of the tax forms used in production as of March 2016. Additionally, I did various amounts of work on about 10 other forms, ranging from bug fixes to implementing significant parts of those forms. The entire application at the time had about 25 tax forms. Implementing a tax form includes doing a web form, xml import/export, PDF reporting, extensive server-side validation based on business and legal rules, and tax form specific changes to the data access layer.

Web application used for submission of data related to various types of location and building permits, built for Croatia's Ministry of Construction and Physical Planning, to be used by general population:

- Tech used: C#, ASP.NET MVC, jQuery, Fluent Validation, Aspose.Words, Oracle Database.
- My role: I did a large part of the entire project: views, models, controllers, business logic, the entire data access layer and PDF reporting.

I worked on frontend/backend development and maintenance of two large ASP.NET Web Forms applications used by the employees of Croatia's Ministry of Construction and Physical Planning.

Created several WCF and ASMX web services as part of the above projects or as standalone components in a larger system.

Created several .NET COM assemblies that are used by legacy applications for communication with other systems, for service calls, authentication etc.

Created various software components used for xml processing: validation, data manipulation, export from database to xml, import from xml to database etc.

Created various programs that do background jobs, such as data processing and validation, email error reporting etc.

Gemmeus d.o.o., Remote

October 2012 - January 2013

C# Developer

.NET component that enables programmatic manipulation of spreadsheet data and conversion to various data formats (.xls, .xlsx, .ods, .pdf etc.):

- Tech used: C#, Web Forms, WPF, Windows Forms.
- My role: Various development tasks. Example: implementation of export from Excel format to a Windows Forms DataGrid (data, formats and styles).

Inetec d.o.o., Zagreb (Croatia)

November 2009 - October 2012

C# Windows Desktop (WPF) Developer

Large WPF desktop application used for damage analysis in steam generators of nuclear power plants:

- Tech used: C#, WPF, Cinch MVVM Framework.
- My role: I developed a large part of the entire application. This includes: several custom WPF controls which display 2D and 3D signal charts/plots (and do the rendering work on a separate thread to make the application more responsive), data persistence layer and data versioning system, app subsystem for signal filtering and mixing, and in general many of the WPF views and view models and the underlying code down to the persistence layer.

WPF application used for observation and control of a robot (prototype) used for probe positioning inside a steam generator of nuclear power plants:

- Tech used: C#, WPF, Cinch MVVM Framework.
- My role: I implemented most of the application. I created all of the GUI code, which mostly consists of custom controls with specialized graphical display. Again, rendering in some of the more complex controls was done in a separate thread to make the application more responsive. I also either created or at a certain point completely overhauled nearly all of the backend code (down to the electronics controller interface), as well as implemented changes and fixes in the C-like code of the electronics controller.

I've also done development and maintenance on several other projects used in other parts of inspection processes in nuclear power plants.

Ultra d.o.o., Karlovac (Croatia)
C# Windows Desktop and Web Developer

October 2008 - November 2009

Web application used for configuration and control of fuel dispenser drivers at fuel stations:

- Tech used: C#, ASP.NET Web Forms, PostgreSQL.
- My role: I implemented some of the project's subsystems and was responsible for the maintenance of the project.

Desktop application used for control and oversight of fuel dispensers at fuel stations:

- Tech used: C#, Windows Forms, SQLite.
- My role: I created the entire application. The application had custom GUI controls that displayed fuel dispenser state and enabled control over fuel dispensers. Data was persisted in an SQLite database.

I also worked on fuel dispenser drivers and other small programs used for regex validation, manipulation of registry entries etc.

Ekobit d.o.o., Zagreb (Croatia)
C# Web Developer

February 2008 - October 2008

I worked on a few application for various financial institutions (banks, funds, stock exchanges):

- Tech used: C#, ASP.NET Web Forms, Microsoft SQL Server.
- My role: I did relatively small tasks and bug fixes on the application and database.

Visage Technologies (Student project), Zagreb (Croatia)
C++ Developer

January 2007 - August 2007

Body animation exporter, part of a larger face and body animation system.

- Tech used: C++, 3ds Max SDK.
- My role: I implemented the entire body animation exporter which converted 3ds Max Biped format to a format called MPEG-4 FBA.

EDUCATION

Faculty of Electrical Engineering and Computing (FER), University of Zagreb
Graduate Engineer of Computing - five year course, total 284 ECTS credits.

SKILLS OVERVIEW

Recent and Substantial Experience:

Languages	Web Frontend Languages (JavaScript/HTML/CSS) and variations (SCSS etc.), TypeScript
Tech/Tools	React, Angular, Node, Express, NestJS, ESLint, Prettier, NX (monorepo), RxJS, Jest, React Testing Library, Webpack, Storybook, Redux, redux-observable, Material-UI (UI component library for React), Zod (validation), NPM (obviously)
Databases	Mongo, PostgreSQL
CI/CD	Github Actions, Heroku
Source Control	Git
IDEs	Webstorm, IntelliJ IDEA

Previous Experience:

Languages	Kotlin, Java, C#, C++
Tech/Tools	Spring Boot, JPA/Hibernate, Liquibase, Maven, Vue, Vuetify (UI component library for Vue), AngularJS, JUnit, ASP.NET Web Forms, ASP.NET MVC, WPF, Windows Forms
Databases	H2, SQLite, Microsoft SQL Server
CI/CD	Github Actions, Heroku
Source Control	TFS, SVN, Jazz (IBM)
IDEs	Visual Studio, Eclipse

Limited Experience:

Languages	Python, Ruby, F#, R, Octave
Tech/Tools	Cypress, GraphQL, Android SDK, libGDX, Docker, Docker Swarm
Databases	MySQL, IBM Db2, Oracle
CI/CD	CircleCI
IDEs	Visual Studio Code

PERSONAL INFO

Name	Goran Mržljak
Gender	Male
Location	Zagreb, Croatia
Nationality	Croatian
Languages	Croatian (native) and English (fluent)
Other	I have a category B driver's license.