



TATIANA KARAMORINA

Frontend Developer

CONTACT

Email: karamorina@gmail.com

Location: London, UK

GitHub: [tappiola](https://github.com/tappiola)

LinkedIn: [tatiana-karamorina](https://www.linkedin.com/in/tatiana-karamorina)

Portfolio: <https://tappiola.com/>

LANGUAGES

English: C1 (Advanced)

Russian: C2 (Native)

EDUCATION

Bachelor's degree in World Economics

CORE FRONTEND SKILLS

HTML, CSS, JavaScript, React, Redux, TypeScript, Svelte

REACT FEATURES

React hooks
Class components
Lazy Loading
React Context
HOCs

REACT LIBS

react-router
react-hook-form
react-testing-library
redux-toolkit
react-transition-group

JAVASCRIPT FEATURES / LIBS

Storybook
Growthbook
Carbon components
Axios
Chart.js
Mapbox GL
PropTypes
Apollo GraphQL
Frontegg
clsx

Tatiana is a frontend developer proficient in modern frameworks such as ReactJS and Svelte. 8 years of prior experience in a related domain (QA) give her extensive technical background and ability to easily predict potential issues during development. She has basic backend development skills. Tatiana is a quick learner and able to work in rapidly changing start-up environments efficiently. She is comfortable working in small teams, not afraid to take responsibility and perform a broad scope of tasks. Tatiana is always diving deep into project details and architecture and constantly suggests the ways to improve the project (from code quality to UX and processes).

WORK EXPERIENCE

FRONTEND DEVELOPER (React + TypeScript)

Ondat, Jun 22 – Jul 22 (London, UK)

Development of a portal for monitoring and configuring Kubernetes clusters.

FE stack: React, React hooks, React Router v6, react-query, react-hook-form, styled-components, carbon-components, Typescript, GraphQL, Jest, React Testing Library, Frontegg, plop, Storybook, Growthbook, Github Actions, dependabot.

BE stack: Nest.js, Prisma, Kubernetes, Docker, Pub / Sub, Redis, PostgreSQL, JWT.

Work done:

- Developing new pages and components according to Figma designs
- Participating in defining requirements and design for new features
- Refactoring existing code, bringing everything to a common standard
- Setting up communication with BE using GraphQL queries/mutations
- Optimising components' structure, adding plop templates
- Developing new forms with complex validation using react-hook-form
- Upgrading existing packages to newer versions, integrating dependabot into Github actions for automatic dependency upgrades
- Investigating new tools and ways to improve the application's performance
- Creating reusable UI components, optimising styled-components theme
- Developing unit tests and StoryBook components

Achievements:

- First new feature developed and delivered during the first week of work
- Test coverage increased by 10% in a month
- Encouraged the team to start developing new features supporting responsive design. Reworked core components to scale on mobile devices
- Improved architecture of multiple components (rewrote modals, improved usage of React Router, added more React hooks, etc.).

CSS TOOLING

SCSS, styled-components, CSS modules, Tailwind CSS, media-queries, flex layout, grid layout

LINTERS

ESLint, StyleLint

CODE TRANSFORMATIONS

Webpack, Babel

BROWSER TOOLS

Chrome / React / Redux / Apollo DevTools

UNIT TESTING

Jest, Enzyme, react-testing-library, Cypress

E2E TESTING

Selenium WebDriver, pytest, Allure Reporting

PERFORMANCE TESTING

Lighthouse, JMeter

BACKEND SKILLS

Node.js basics, Nest.js, Prisma, Sequelize, Express, Express-graphql, Firebase, JWT

DATABASES

SQL (MySQL, PostgreSQL, etc.), NoSQL (Redis, MongoDB, Firestore)

API

REST, JSON, Fiddler, Postman, Altair GraphQL Client

VERSION CONTROL

Bitbucket, GitHub

DEPLOYMENT TOOLS

TeamCity, Jenkins, Docker, AWS, Azure

FRONTEND DEVELOPER (React)

Scandiweb, Feb 21 – Jan 22 (Riga, Latvia)

Development of ScandiPWA Core – modern React-based frontend app, designed to fully replace out-of-date Magento themes. Working as part of international team.

Tech stack: React, Redux, SASS, GraphQL, Magento, PWA.

Responsibilities:

- Bug fixes
- Implementing support for Magento features (i.e., frontend customizations for different CMS settings)
- Developing theme extensions (i.e., packages for replacing whole features)
- Developing complex UI components with custom animation logic (e.g., carousel, swipe to delete, fullscreen image zoom, image gallery)
- Refactoring, improving performance, error handling, loading behaviour
- Adding new GraphQL endpoints, implementing minor backend changes

Achievements:

- Redesigned the whole website according to new Figma designs
- Improved lighthouse scores to 90%+
- Added multiple custom ESLint rules + StyleLint to improve code quality
- Improve SEO by adjusting server-side rendering logic
- Added RTL support to components

Demo website: <https://tech-demo.scandipwa.com/>

Here you can access the project's [source code](#) as well as Tatiana's [contributions](#).

QA ENGINEER

Easybrain, Dec 18 – Aug 20 (Minsk, Belarus)

Proud to contribute to the financial results of the developer of the most popular [Sudoku](#) app in the world and many other mobile apps. Was a QA engineer of a big data application designed to accumulate and analyse marketing and financial statistics. Tasks included testing of the reporting portal, ETL processes, Tableau reports. Tatiana was the QA engineer who introduced tests automation on the project (using python and pytest), which turned out to be very effective considering big data project specifics. She contributed to formalizing the development flow, which resulted in quicker features delivery and less bugs in production.

QA ENGINEER

Wargaming.net, Nov 17 – Oct 18 (Minsk, Belarus)

Testing of a React-based e-commerce app integrated into [World of Tanks](#) client (ingame shop), releases, test environments setup and support. Carrying on the app's initial release. Tests automation (pytest), working with mocks, Docker, TeamCity.

QA ENGINEER

A1QA, Jul 12 - Oct 17 (Minsk, Belarus)

Working on multiple mobile and web projects as Manual and Automation QA Engineer. Tests automation mainly using python and Selenium WebDriver. Working as part of multinational teams, including ones based in the UK (e.g., [Product Madness](#)). Tatiana was the first QA engineer ever of the famous [Heart of Vegas](#) game.

PET PROJECTS

REAL ESTATE AGENCY WEBSITE (React + TypeScript)

Web app allowing to search for real estate properties, filter them and view on interactive map.

Selected features: user authentication, animations, responsive design, dynamic list loading, react lazy loading, custom carousel, notification stack, images gallery, skeleton loaders.

Tech stack: React, React Router v6, redux-toolkit, SASS, Typescript, react-transition-library, Express, GraphQL, sequelize, JWT, Docker, docker-compose.

Links to [live website](#) and [source code](#).

TASK MANAGEMENT APPLICATION (React)

Web app for creating and managing todo lists. Tasks completion, editing, removal and moving them between folders is supported.

Tech stack: React, React Router v5, React Hooks, Redux, Firestore, Firebase Auth, Jest, Enzyme, prop-types.

Links to [live website](#) and [source code](#).

ONLINE GAME (React)

Multiplayer game for guessing words by making associations to a keyword.

Features: pseudo-random game fields generation, complex game progress and winner detection logic, webSockets, multiple languages, mobile devices support.

Tech stack: React, React Router, React Hooks, React Context, Firestore, clsx.

Links to [live website](#) and [source code](#).

E-COMMERCE SITE (React)

E-commerce website with products search, catalogue, cart and checkout.

Tech stack: React, React Router, Redux, axios, Django Rest Framework, PostgreSQL

Links to [live website](#) and [source code](#).

STOCKS MONITORING APP (Svelte)

App for viewing statistics on the shares of your choice for the last trading day.

Tech stack: Svelte, Chart.js, axios.

Links to [live website](#) and [source code](#).

WORD GAME (Svelte)

Analogue of a famous Wordle game. Aim of the game is to guess a word by info on presence of letters in a word and their positions.

Links to [live website](#) and [source code](#).