

Bekir Ugur

Software Developer



As a +2 years Certified Software Developer, having the ability to learn quickly, attention to details, adaptation to teamwork and being self-motivated play an active role in projects. Offering combination of proven problem-solving, analytical thinking, adaptation and communication skills. Seeking to join a synergic and harmonious team as a Full Stack Developer to develop amazing applications.

Contact

Address : Konya / Turkey

Phone : +90 542 648 79 96

E-mail : bekirugur070@gmail.com

Github : [www.github.com/bekirugurr](https://github.com/bekirugurr)

Linkedin: www.linkedin.com/in/bekirugurr/

Portfolio site: <https://bekirugur.netlify.app/>

Software

Frontend:

JavaScript | Typescript | React.js | Next.js |
Redux | HTML | CSS | Tailwind | Bootstrap
| SASS | Material UI | Jest | Testing Library

Backend:

Python | Django | Restful APIs
SQL | PostgreSQL | OOP

Others:

Git | Github | Linux
Agile | Jira | SDLC

Want to Learn:

React Native | Docker | AWS

Soft Skills

Communication | Collaboration
Teamwork | Responsibility
Adaptability | Empathy
Analytical Thinking
Problem Solving
Time Management
Lifelong Learning

Languages

Turkish (Native)

English (Advanced)

Experience

2022-07 - 2022-10 **Freelance Developer Remote**

- Developed functional and sustainable web applications with clean codes.
- Made designing, developing, debugging, testing, documenting and deploying. Used Agile methodologies (Scrum and Jira).

Education

2022-9- 2022-9 **Frontend Developer Patika.Dev Practicum – Turkey Remote**

- In this practicum, typescript and next.js learned. A popup-generator app developed that provides fully customizable popups. By this app, users can get different popups to use in their sites without knowing coding.

2021-09 - 2022-07 **Full Stack Development Certificate Program Clarusway IT School - USA**

- Successfully graduated after 9 months of intensive work. Attended 180 hours for IT fundamental, 240 hours for Frontend and 120 hours for Backend, in total 540 hours of in-class lessons. Completed 1000+ hours of pre-class and post-class activities.
- Completed 20 middle-level and 1 capstone hands-on project.

2008-09 - 2012-06 **Faculty of Security Science (Bachelor's Degree) Turkey**

Certifications

2022-10 - Testing React with Jest and React Testing Library (Udemy)

2022-08 - Javascript (Sololearn) and jQuery (Sololearn)

2022-07 - Fullstack Course Certificate (Clarusway)

2022-07 - Backend & Django Course Certificate (Clarusway)

2022-05 - Frontend & React Course Certificate (Clarusway)

2022-05 - HTML/CSS/Javascript Course Certificate (Clarusway)

Fullstack Projects

- **React Blog App** : It was aimed to create blog posts and display them on the dashboard. Register, login, logout and updating profile can be done. Read only users just can see dashboard and 3 initial lines of post. Authenticated users can add new posts, see details of all posts, like or dislike posts and make comment to posts. Writer of a post can update or delete this post. Number of unique views of posts were store and display in each detail page (Repeated views are not count). By pagination, when every click to 'view more', 6 posts are added to dashboard.

Frontend Skills/Tools : React, Material-UI Library, Axios, Redux, React-router DOM, Formik and Yup Libraries, React-toastify Library. **Backend Skills/Tools** : Django Rest Framework, PostgreSQL, Heroku, Concrete views, Viewsets, Token authentication, custom permissions, object permissions, cursor pagination, dj rest auth, nested serializer, swagger, redoc, debug toolbar.

Frontend Projects

- **React Popup Generator**: It creates modular popups that customer can use them into their sites without knowing coding. Customers can customize size, content, position, color of popup. They can add image if selected popup allows. They can determine in which kind of device (desktop, mobile or both) popup will appear, how many seconds after site opened or after what percentage of the page scrolled the popup will appear. Customers can also allow to popup to appear for whom only come with links from certain sites and who have certain browser languages. Finally customers can allow popup to appear when user moves out the mouse cursor to close tab. After these customization process, customers get a short code snippet. After just pasting this snippet to a body of an html file the popup will work. Only requirement is tailwind must be installed to the html. **Skills/Tools** : Next js, Typescript, React, Vanilla Javascript, Tailwind, Redux Toolkit.
- **React Movie App** : It displays film details, posters and fragmans by using the tmdb api. There is a search feature according to the entered word. Read only users just can see overviews and posters of films in dashboard. Authenticated users can see details of films. In addition, Firebase was used for register, login and authentication. Context API was used. **Skills/Tools** : React-router library, Bootstrap library, Context API, Axios and Firebase
- **React Personal Portfolio**: Fully responsive personel website. Framer Motion and react_scroll are used for animation and transition. Webhook was used for receiving messages. **Skills/Tools** : React, Framer Motion, Tailwind, Webhook.
- **Checkout Page** : A real e-commerce website's checkout page. There is a cart for buying products, and we can increase, decrease the number of products or remove the product. By changing a parameter in the cart, our js functions calculate the cart total price. **Skills/Tools** : DOM, Iterator, Callback functions, String-array functions and Events.

Backend Projects

- **Django Quiz API** : It generates test quizzes. There are answers under questions under quizzes under categories. Admin panel was customized as nested admin panel so that the admin can enter the exam, questions and answers together. Adding, deleting and updating can be done. **Skills/Tools** : Django Rest Framework, PostgreSQL, Concrete views, Token authentication, custom permissions, dj rest auth, nested serializer, swagger, redoc, debug toolbar.
- **Flight Reservation API** : An API that allows create, update, delete or read flights, reservations and passengers in different ways according to whether user is staff or not. Authenticated ordinary users can create, update, delete or read reservations. Only admin users can add, update, delete or read flights. While admin user can see all reservations authenticated ordinary users can see the reservations that they made. While admin users can see flights with their reservations others can only flights informations. User can register, login, logout, update profile. **Skills/Tools** : Django Rest Framework, PostgreSQL, Concrete views, Token authentication, custom permissions, dj rest auth, nested serializer, swagger, redoc, debug toolbar.