

İsmail Bilge Önceken

Yeşilbayır Mah. 1920 Sokak 6/7
Mamak/Ankara TURKEY
onceken.bilge@gmail.com
(+90)506-225-5195

Education

Middle East Technical University

2014 September - Present

BSc in Mining Engineering

Senior

Ankara Atatürk High School

2008 September - 2013 June

- One year English preparatory school
- Taken Chinese as secondary foreign language

Work Experience

Software Related:

Jotform

2018 June

Web Development Intern

Javascript/React

Mining Related:

Koç Demir Export Divrigi Iron Mine

2018 August

Mining Engineering Intern

Eczacıbaşı Esan Balya Underground Lead-Zinc Mine

2017 August

Underground Mining Engineering Intern

Voluntary Work

Orienteering Instructor at METU Summer Sports School 2015

2015 July

Taught orienteering and planned indoor/outdoor
orienteering parkours for 8-12 age group.

Volunteer at İznik Ultra 2015 Ultra Marathon

2015 April

Set up and managed "Narlica 90k Control Point" with a team of six.

Honours & Awards

HAVELSAN GIS Solution Competition 2nd Place

2017 December

Implemented a Quantized-Mesh encoder in Java in order to present raster elevation data with Quantized Mesh protocol developed by the firm Analytical Graphics. Technical challenges were related with geographical calculations, triangulation, file structure and data visualization. Used general web development skills to cover incomplete parts of the submitted solution.

Problem Definition

Win Proof

Başkent University Ekin Pre-Incubation Center
Entrepreneurship Contest 3rd Place

2017 June

Developed a business idea with a group of four.
Made a pitching presentation to a jury of academics and business professionals.
Later, the same project has been chosen as "Successful Business Idea" and provided with mentorship support from Ankara Development Agency.

Result Announcement

Side Projects

react-suspense-image-loader

A progressive image loader component for react.

Uses unstable react suspense technology which brings a new asynchronous rendering mechanism for UIs. Uses [react-transition-group](#) for providing easily customizable load animations and placeholders.

For a smooth web experience for our users we needed to reserve space for the image to be loaded and give it some time to load in order to prevent flickering. If not quick enough, then we may want to use spinners or blurred placeholders. react-suspense-image-loader provides easy solution for such situations.



[Code Sandbox: https://codesandbox.io/s/j406rjr1ow](https://codesandbox.io/s/j406rjr1ow)



[Github: https://github.com/bilgeoncken/react-logging-error-boundary](https://github.com/bilgeoncken/react-logging-error-boundary)

react-logging-error-boundary

A logging error boundary component for react which logs cached errors to [Sentry](#) and/or [Loggly](#).

Just provide credentials for your services. They will be initialized and starts collecting thrown errors.



[Code Sandbox: https://codesandbox.io/s/9z74q1lm14](https://codesandbox.io/s/9z74q1lm14)



[Github: github.com/bilgeoncken/origames](https://github.com/bilgeoncken/origames)

Origames - Fantasy Orienteering League Game

A web application built for METU Orienteering and Navigation Team.

Built with [Django](#) and [MaterializeCSS](#). Served with [Nginx](#) on a [DigitalOcean](#) droplet.

For every official race, users create their teams by picking up 9 members of METU Orienteering team. Scores are calculated automatically after the finish times are entered for the orienteers.

At the end of the race period the team with the highest score wins.

Live at: <https://origames.org> Username: demo Password: Demo1234



[Github: github.com/bilgeoncken/origames](https://github.com/bilgeoncken/origames)

Skorbela - Score-O Result System

A web application built for Dec. 2015 - New Year Score-Orienteering Event organized by METU Orienteering and Navigation Team. Abandoned since the team now uses [SportIdent](#) and its software for organizations.

Built with Asyncio-based [Autobahn](#)/[Python WebSocket Framework](#) and [Bootstrap](#) without a prior knowledge of web development. Uses a JSON file as database via [jsondatabase](#).

Initial aim for the application was to remove the messy process of five people entering scores of many registered competitors to excel on five different computers for one race. To enable users to edit json file simultaneously, We've decided to use Autobahn|Python which let us create a websocket server working fully asynchronously.

Users can add new control points, competitors and enter competitors' results easily.

No live demo for this one because I did not know about requirements.txt back then.



[Github: github.com/bilgeoncken/Skorbela](https://github.com/bilgeoncken/Skorbela)

Skills

Programming Languages and Tools

- Python, Javascript, Java, HTML, CSS,
- Django, Flask, React
- PostgreSQL, ORMs, GNU/Linux, Git, Heroku,



github.com/bilgeoncken

Spoken Languages:

Turkish
Native

English
Full Professional
Proficiency

Mandarin Chinese
Beginner

Interests

- Orienteering at METU Orienteering and Navigation Team
- Programming
- Trail Running
- Electric Guitar