

Apostol Gabriel

Frontend • Backend • UI/UX

0761030884 | apostol.gabriel90@gmail.com | My website | LinkedIn | Github

Work & Education

Javascript developer

05.2019 - 05.2021

Advahoo, Bucharest (Romania)

- Mostly work on front-end and UI/UX, occasionally assisting backend and our clients in Testing, overall using Agile methodology
- Used Oracle jet and React to build and maintain features of a complex Single page applications for PC and Mobile
- Participate in building the app's Architecture for best Modularity, Performance, and Adaptability in concordance with customer needs

Full stack (java and frontend) - Internship

3 months

Unicredit Services, Bucharest (Romania)

- Worked on the Frontend and UI/UX on a project made for a particular group of clients. Providing and implementing solutions for their needs.
- Implement Backend APIs in JAVA in concordance with Business logics

Bachelor degree in Computer Science

3 years

University of Bucharest, Bucharest (Romania)

Skills

Worked most with

REACT VUE NODEJS JAVASCRIPT OJET CSS3
HTML5 MYSQL RUBY ON RAILS

Technologies and Concepts I worked with

MONGODB CONTINUOUS INTEGRATION (CI) JIRA
UNIT TESTING JEST EXPRESS LINUX PGSQL
GITHUB NPM SCSS LESS PYTHON DJANGO
UIKIT BOOTSTRAP MVC REST OOP AGILE AJAX
JQUERY REDUX WORDPRESS

Projects

Personal trainer site and blog

A blog and a custom personal page for a personal trainer, made in express and mongodb. The trainer can login and edit articles, add toturials with images and videos. Each on its own section. The site is simple designed and lite coded destined to be simple to read and to run fast. All the dynamic content that the trainer is added is updated on cloudinary using the cloudinary API.

- Used database: MongoDB, cloudinary
- Frontend tool kits: ui-kit3, jquery, less
- Backend: Express, EJS

Skyline

Modular article manager in Ruby on Rails that supports database replication using Octopus gem, paragraph editing, photo galleries, user profile, search, and tags for each article. In addition to this project, I made a fully functional replication topology that worked with this application.

- Used database: MySQL
- Frontend tool kits: ui-kit3, jquery, less
- Backend: Ruby on Rails