Curriculum Vitae
Iurea Marius

marius.iurea@yahoo.com — 0741055112 — (Skype)marius.iurea1

Personal Profile

Dedicated, self-motivated, and passionate student, looking to fill a position where I can learn and take on new challenges. My main goal is to experience the whole software development cycle and enhance my abilities in designing, implementing, testing, maintaining and upgrading software. I am confident that if given the opportunity, I can be a useful part of your software development team. This CV is under source control, get the latest version at: (Updated CV)

Technical Skills

- **Languages** — C, C++, C#, Python, HTML, JavaScript

- **Frameworks/Libraries** — ASP.NET, JQuery, Django

- **Tools** — Git, Unity, Visual Studio, Vim

Projects

- **RedditFeud** — Webapp built using Django, hosted on AWS (Link)

  Motivation:

  As an avid user of reddit, I wanted to write a webapp that would interface with the python reddit API and since python seemed like an easy to use language I decided to learn Django. Another important part of this project that I tried to give special attention was going through the process of buying a domain and hosting because I had no prior experience doing it.

  Summary:

  This webapp queries the python reddit API and stores the most popular posts and comments submitted today in an SQLite database, then when a user requests a quiz it pulls 10 randomly chosen questions from the database and uses the template engine to create the webpage.

  Skills Acquired:

  -The ability to plan out a bigger project and go through the steps required to turn an idea into a user faced product.

  -Understanding of how a web framework deals with receiving, parsing and responding to HTTP requests using the MVC design pattern.

  -Knowledge of how to use an API in order to gather and serialize data in order to more easily retrieve it when it is requested by the user. Also got experience in finding my way

through documentation.

-Processing of JSON files was a key part since the API had a pretty restrictive rate limiter I had to use a third party API which compiled and archived bigger chunks of posts and comments.

-Experience using a template engine in order to dynamically fill out webpages.

-Ability to use CSS Grid to resize and scale the content of the webpage for different devices, JQuery was used for user interaction.

-Basic storage of posts and comments in an SQLite database in order to make adding them to the templates easier.

-Usage of git in order to manage different versions of the project, sync up on multiple devices, and keep a ledger of changes I made in order to more easily revert them.

-The process of deploying my project to AWS using Nginx and setting up daily cronjobs to keep the database updated with new content daily.

- **Unpopular Opinions** — Website built in ASP.NET with a Bootstrap front end for FiiPractic full-stack ASP.NET courses (Link)

  Motivation:
  Nearing the end of these courses we were tasked with putting our newly acquired knowledge to use, we were told to build whatever we wanted as long as we used everything we were taught in those 5 weeks.

  Summary:
  Message board website that stores users posts and comments, has a reply, up and downvote systems for sorting content, authentication with a Google account, and API support. It was one of the three winning projects.

  Skills Acquired:
  -Learning a framework in a relatively short amount of time and putting it to use under a tight deadline.

  -First-hand experience with the C# programming language, the only prior experience was a little bit of scripting for a game using Unity.

  -Creating tables and modeling relations between them in the database using the code first approach in Entity Framework in order to create one-to-one, one-to-many and many-to-many relationships.

  -Razor template engine in order to fill out templates on demand to be shown to the user.

  -Further developed my understanding of the MVC design pattern and routing.

  -Got more hands-on experience with the different types of HTTP requests and error codes.

  -Developed an API and used it in order to dynamically send requests and receive JSON files using AJAX and JQuery.

  -Used Microsoft SQL Server.

  -Understood how model binding retrieves data from form fields, query strings and route data, converting string data to .NET types.

-Authenticating users using their Google account.

-Learned how to use LINQ in order to more easily retrieve objects from the database in formats that are easy to manipulate.

-Used Bootstrap to quickly develop the frontend of the website.

-Learned how to use data annotations in order to validate user input and return appropriate error messages.

- **Bonol** — Implementation of the L game in C++ for a project in university (Link)

  Motivation:
  This project was built in collaboration with a colleague of mine for the "Introduction to Programming" course. Using any sort of object oriented programming was forbidden.

  Summary:
  The L Game is a simple abstract strategy board game. The game was designed to be difficult and skill expressive without having many board pieces. The game has a vs CPU mode as well.

  Skills Acquired:
  -This project helped me develop my problem solving abilities and my knowledge of C++.

  -I got better at working in a team.

  -Had to develop a GUI with a barebones graphics library.

  -Used git for version control and cooperation.

Education

- **Currently getting a bachelor's degree at the Faculty of Computer Science Iasi(first year student)** — 8.66/10 Grade Average

  Got into the university with a 9.85 grade on the admission test.

  Throughout highschool, I participated in the informatics olympiad and was interested in competitive programming.

  At the end of 12th grade participated in Oracle SQL courses.

  Favourite Courses:
  -Computer Architecture and Operating System was my favourite course in the first semester, mostly because I found it to be a lot more difficult than the other subjects. It was something I hadn't worked with before and it sparked my interest in how computers work on a lower level.

  -Object Oriented Programming was my favourite course in the second semester, because when compared with other courses, it really felt useful and I could see how and why I would use what I was learning.

- **Winning project in FIIPractic full-stack .Net courses** — The technologies we used were ASP.NET, Microsoft SQL Server, HTML, CSS, Bootstrap, Javascript, JQuery, AJAX, and Git.