Denisa Predescu

Nationality: Romanian **Date of birth:** 08/02/2001

☎ **Phone number:** (+40) 737603428

✉ **Email address:** predescu.denisa2001@gmail.com

Home: Str Mircea Marinescu nr 35B , 077191 Voluntari (Romania)

WORK EXPERIENCE

Software Engineer Intern

Adore Me [07/2022 – 10/2022]

Address: Intrarea Rigas 31 bis, 010152 Bucharest (Romania)

I was introduced to the *PHP* programming language and to the *Laravel* framework.
I learned to use the REST API; use the controller-service-repository structure required in the backend; create, modify and work with databases; create, push and consume queues;

Junior Developer Intern

PHINIA IT Services [01/2024 – 07/2024]

City: Bucharest | **Country:** Romania

As an intern I was exposed to different technologies like *Python, Laravel, PHP, Power Platforms* (*Power Automate* and *Power Apps*).
I also learned to work with *SharePoint* sites: libraries, lists, permissions, integrate custom *Power Apps* and *Power Automate* flows

Developer - Corporate Web Applications

PHINIA IT Services [07/2024 – Current]

City: Bucharest | **Country:** Romania

Continue to work with *low code, no code* technologies like *Power Platforms, SharePoint,* but also extend to *PowerShell scripts, HTML* and *CSS* projects

EDUCATION AND TRAINING

Highschool Diploma

National College 'Cantemir Voda' [2016 – 2020]

Address: Strada Viitorului 60, 030167 Bucharest (Romania) | **Website:** https://www.cncv.ro/

Bachelor's Degree

University of Bucharest, The Faculty of Mathematics and Computer Science [2020 – 2023]

Address: Str Academiei 14, 010014 Bucharest (Romania) | **Website:** https://fmi.unibuc.ro | **Field(s) of study:** Computer Science | **Final grade:** 9.69

Master's Degree

University of Bucharest, The Faculty of Mathematics and Computer Science [2023 – Current]

Address: Str Academiei 14, 010014 Bucharest (Romania) | **Website:** https://fmi.unibuc.ro | **Field(s) of study:** Database and Software Technologies

LANGUAGE SKILLS

Mother tongue(s): Romanian

Other language(s):

English

LISTENING B2 **READING** B2 **WRITING** B2

SPOKEN PRODUCTION B2 **SPOKEN INTERACTION** B2

Levels: A1 and A2: Basic user; B1 and B2: Independent user; C1 and C2: Proficient user

DIGITAL SKILLS

Good Knowledge

Power platform (Power Apps, Power Automate) / SharePoint Online/O365 / Python / Javascript, TypeScript / PL\SQL / SQL, Oracle / C++ / Object-Oriented Programming / HTML / CSS / c# / php / Java / Microsoft SQL Server (SSMS, SSIS) / Angular Js / PHP Framework Laravel / Shell Scripting (Powershell, bash,..)

Beginner

C / Node.js / Haskell / R

Also used

Linux / Postman / Github / XML / MySQL / RESTful api

PROJECTS

Veterinary office - OOP Management of a veterinary office, done in a *C++* terminal using *object oriented programming* elements and structures. The program was able to manage customers as well as employees and bills. It was able to discount customers that brought the same pet more than once, it was able to manage pay increases for employees, print bills and much more.
Programming languages used: *C++*.
(Made for Object Oriented Programming course in Year 1)

Link: https://github.com/denisapredescu/Veterinary-office

Website presenting the city of Suceava - Web Application A informational website using *HTML*, *CSS*, *JavaScript* and *Node.js*, about the city Suceava. The website contains information about historic, touristic and geographic data alongside general information. It also contained a section for users to add pictures and share their experience.
Programming languages used: *HTML, CSS, JavaScript, Node.js.*
(Made for Web Techniques course in Year 1)

Link: https://github.com/denisapredescu/TehniciWeb-Prezentare-Suceava

Library Management - PL/SQL Developer A *PL/SQL* project that required making a database and it's tables to manage a library. I made commands on tables to filter and select data included. The project had at least 10 tables and each one at least 10 entries for each one. It had type 2 and 3 relationships between tables and had all the basic *S QL* commands such as *CREATE*, *UPDATE*, *INSERT*, *DELETE*.
(Made for Database Management Systems course in Year 2)

Link: https://github.com/denisapredescu/Library-Management

Shell Command Line - C The Program mimics a shell terminal, we implemented the base commands as : *pipe*, *cd*, *cat*, *logic operations*, *pwd, history*(who returns a list of the executed commands*)*, *mkDir*, *echo , etc.*
The user is also known. The app is capable of opening and executing programs.
(Made for Operating System course in Year 2)

Link: https://www.dropbox.com/s/tlfgsi5xaitliaf/ProiectSO_Predescu_Postolache_SanduR.zip?dl=0

Medical Office - Java Management of a medical office, done in a *Java* terminal using *object oriented programming* elements and structures. The program displays a different menu depending on the role (doctor or admin) and the functionalities are specific to the roles. The doctor is able to see information about patients, insert patients and details about them (appointments, tickets, prescriptions). An admin can add and delete doctors, change salaries, view doctors and patients.
The information is stored in a local database (*MySQL Workbench*).
(Made for Advanced Object Oriented Programming course in Year 2)

Link: https://github.com/denisapredescu/CabinetMedical

Bookstore - .NET + Angular A *.NET* web application using *Angular* for frontend elements for a online Bookstore. The project allows you to create an account and see the available books with some information obtained from a local database, it also uses the database to display information about authors and the books they wrote. You can place orders and see the order history.
Programming languages used: *C#*, *Angular*, *HTML*, *CSS*, *TypeScript*.
(Made for Advanced Front-end (Angular and React) course in Year 3)

Links: https://github.com/denisapredescu/Online-Bookstore | https://www.youtube.com/watch?v=Q0AuPsbrTtw

Meteorite shower - Arduino IDE Meteorite shower is a game inspired by the Space Invader game, done in *Arduino IDE*. The main idea of the game is to hit the meteorites that appear random from the upper side of the matrix at an interval of time. Changing the difficulty is basically changing the number of meteorites that come at once, the interval of their appearance, the number of lives of the meteorites and the points gained from destroying them. At the end of the difficulty is a boss fight that is harder to defeat.
In the menu, the user has the possibility to see the leaderboard (saved in *EEPROM*) and reset it, enter his name and modify certain settings such as difficulty, LCD contrast and brightness, matrix brightness, set volume on/off.
Programming languages used: *C++*.
(Made for Introduction to Robotics course in Year 3)

Links: https://github.com/denisapredescu/Meteorite-shower | https://www.youtube.com/watch?v=4IV-Ehv1Mmc

My agenda - Android application A *Java* Android application made in *Android Studio* that function as a booklist agenda. The application allows you to register or sign in with a google account. A user can search for a book in the *se arch page* by entering a title or a part of the title. Using *Google Books APIs*, a list of books and some details are displayed and the user can add one or many to his *to read* list. He can also add them manually by completing some fields and (if he wants) accessing the camera to take a pictures of the book. He can move them from *to read* list to *fini shed reading* list. A notification message is displayed when a new book is added and can redirect the user to the certain page.
Programming languages used: *Java*.
(Made for Programming Mobile Devices in Android course in Year 3)

Link: https://github.com/denisapredescu/My-agenda

TSP App - Traveling salesman Problem Web Application TSP App is a web application built using *Angular* and *Flask*. The application is informative, discussing *The traveling salesman problem* and some algorithms to solve it. Algorithms are presented schematically along with images, step by step examples, and pseudocode. Algorithms can be run in the application on different types of inputs that the user can choose from.
Programming languages used: *HTML*, *CSS*, *TypeScript* (in the *Angular* framework) and *Python* required for *Flask*. The application uses some special modules to display information like *SigmaJs* and *Graphology* to create dynamic graphs.
(Made for the practical part of the Bachelor's degree)

Links: https://github.com/denisapredescu/TSP-App | https://github.com/denisapredescu/TSP-algoritmi