|  |
| --- |
| **Candidate Application** |

Please submit CV containing no more than 5 pages in .PDF format

|  |  |
| --- | --- |
| **Supplier** | |
| **Supplier name** |  |
| **Contact person** |  |
| **Phone number** |  |
| **Email** |  |
| **Candidate Vs Position** | |
| *{{ fitness }}* | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Candidate** | | | | |
| **Last name** | *{{ name }}* | | | |
| **First name** | *{{ lastname }}* | | | |
| **Date of birth** | *{{ birthday }}* | | | |
| **Nationality** | *{{ nationality }}* | | | |
| **Gender** | *{{ gender }}* | | | |
| **Employee or Freelance** | *{{ work\_occupation }}* | | | |
| **Availability** | *{{ availability }}* | | | |
| **Professional References** | | | | |
| **Company** | | **Contact name** | **Professional relationships type** | **Contact details** |
|  | |  |  |  |
|  | |  |  |  |
|  | |  |  |  |
| **Education** | | | | |
| {{ formatted\_education }} | | | | |
| **Certifications** | | | | |
| {{ formatted\_certification }} | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Languages** | | | |
|  | **French** | **Dutch** | **English** |
| **Spoken** | {{ french\_spoken }} | {{ dutch\_spoken }} | {{ english\_spoken }} |
| **Written** | {{ french\_written }} | {{ dutch\_written }} | {{ english\_written }} |
| **Comprehension** | {{ french\_comprehension }} | {{ dutch\_comprehension }} | {{ english\_comprehension }} |
|  | *Scale: native - fluent – good - basic* | | |

|  |  |
| --- | --- |
| **Professional experience** *(for every position fulfilled, most recent first, add rows if needed)* | |
| **Company** | Solita |
| **Client** | SOLITA INTERNAL |
| **Period** | 5/2022 - 5/2022 |
| **Role** | DABASE MIGRATION - EXAM SIMULATOR |
| **Tasks** | Migration of data from exam simulator from old to new Database transforming, Flask, cleaning and updating the data to match the new Data Models implemented for the new application version. Technologies: Python, Git, SQL, Microsoft Azure SQL Databases |
| **Tools** | Flask, Git, SQL, Azure |
| **Environment** |  |
| **Methodology** |  |
| Company | Solita |
| Client | BECODE - FEENPOP |
| Period | 9/2021 - 9/2021 |
| Role | DATA ANONYMIZATION |
| Tasks | Data Anonymatizator App to encrypth and anonymaize confidential from SQL Databases from FeenPOP. Technologies: Python, Git, Streamlit , Threading, Mathematics, SQL, Pandas |
| Tools | SQL, Git, Streamlit, Threading, Pandas |
| Environment |  |
| Methodology |  |
| Company | Solita |
| Client | BECODE |
| Period | 6/2021 - 8/2021 |
| Role | BELGIUM REAL STATE PREDICTION API |
| Tasks | The API coded in python to return the predicted price of a properties in Belgium, based on data scrapped from Immoweb from 2021. For the predictions a Linear regression was put in place to compute the relationship between several characteristics found on the sell announcement to estimate of the asking price is made. The accuracy of the model is pf 85%, which means that there is always a possibility for outliers (less then 15 %). This API has been deployed with heroku under the url: https://api-ie-predictions.herokuapp.com/ Technologies: Python, Tensorflow, PyTorch, Git, Scikit-learn, Pandas, Selenium, BeutifulSoup, HTML, Heroku |
| Tools | Tensorflow, PyTorch, Git, Scikit-learn, Pandas |
| Environment |  |
| Methodology |  |
| Company | Solita |
| Client | BECODE |
| Period | 5/2021 - |
| Role | 3D HOUSES VIEWER |
| Tasks | Application coded in python to plot a house in 3D given an address. The data was collected crossing data from LIDAR satelites from Vlaanderen Overheid services and metadata, and geographical data obtained from the address through API’s servies. Technologies: Python, Git, API’s requests, Pandas, Pillow, matplotlib. |
| Tools | Git, Pandas, Pillow, matplotlib |
| Environment |  |
| Methodology |  |
| Company | Solita |
| Client | BUSSINESS&DECISION |
| Period | 11/2021 - 3/2022 |
| Role | EDGE-COMPUTING IMAGE RECOGNITION NETWORK |
| Tasks | Internship project. Create an internetless LAN network capable to enable communication between diferent devices exposing each one as an microservice. Two raspberries were used, first one as camera streaming service and the second as Image Recognition model host, and a phone used as endpoint to check the results of the recognition. Technologies: Python, RaspberryPi, Mimik, Yolov5, Tensorflow, MobileNetV2, RTPM |
| Tools | RaspberryPi, Mimik, Yolov5, Tensorflow, MobileNetV2, RTPM |
| Environment |  |
| Methodology |  |
| Company | Solita |
| Client | BECODE - FAKTION |
| Period | 10/2021 - 10/2021 |
| Role | RETAIL ANOMALY DETECTION |
| Tasks | Application programmed in python to detect anomalies in manufacturing of dices. A Convolution Neuronal Network was trained to classify daces by its face, then a second process function computes the differences between a good manufactured one and the current one, being able to distinguish if the current dice had any anomaly or not. Technologies: Python, Git, CNN, OpenCv |
| Tools | Git |
| Environment |  |
| Methodology |  |
| Company | Solita |
| Client | BECODE |
| Period | 10/2021 - 10/2021 |
| Role | SIGNATURE RECOGNITION |
| Tasks | YOLOv5 model trained in python to detect signatures on documents, it was trained with annotated documents transformed to jpg and addapting their annotations from an .xml format to a .txt normalizing and transpolating the coordinates to the yolo format. Objective: extract signatures from documents to validate legitimacy. Technologies: Python, Git, Yolov5, XML |
| Tools | YOLOv5, Git, XML |
| Environment |  |
| Methodology |  |
| **Management Skills** | |
| *{{ man\_skills }}* | |
| **Technical Skills** | |
| *{{ tech\_skills }}* | |
| **Others** | |
| {{ other\_skills }} | |