PERSONAL DETAILS

Name	:	Rajith Rahul Kumar
Date of Birth	:	24.07.1996
Current Residence	:	Dessauer Straße 65, 70376 Stuttgart
Email	:	rajithrahul24@gmail.com
Phone	:	+49-1768-547-4750
Nationality	:	Indian
Marital Status	:	Married
Professional	:	**MyWebsite**, **LinkedIn**, **GitHub,**

EXPERIENCE

Software Engineer
05/2021-07/2023

Title: Software Engineer – Research and Development
Company: Olmatic GmbH, Freudenstadt, Germany.
Client: Landerer, Oerlikon, Trelleborg, Alzner, Riha Plastic.
Tools: React, Azure, Node JS, SQL, Power BI, Javascript, Typescript, C#, Express, DevOps, HTML, CSS, Sass, Python.

- Developed and maintained scalable web application using Node.js and React.
- Secured authentication was provided by Azure key vaults secrets and B2C services.
- Creative UI/UX were developed for intuitive user interfaces and seamless user experiences.
- Worked with Azure cloud services, deploying and configuring applications for optimal performance and scalability.
- Implemented RESTful APIs with Cosmos DB services for data retrieval, and processing.
- Leveraged Azure cloud services, such as Function App Services, Key Vaults, Cosmos DB, Storage Accounts, Azure B2C and Service Principles, to build serverless applications and automate workflows.
- Designed and optimized databases using Cosmos Db, Stored Procedures and SQL Server.
- Utilized DevOps practices, including CI/CD pipelines to streamline the development process and enhance deployment efficiency.
- Developed interactive data visualizations using Power BI providing insights into complex energy datasets and embedded Power BI reports into web application for customers.
- Assisted in troubleshooting and resolving production issues, ensuring smooth operation of the application.
- Actively participated in agile ceremonies, such as daily stand-ups, sprint planning, and retrospectives, fostering efficient project management and continuous improvement.

Master Thesis
10/2019-08/2020
Grade (1.0/5.0)

Topic: Framework for Advanced Driver Assistance Systems (ADAS) using ROS
Company: AKKA Technologies, Böblingen, Germany.
Client: BMW, Daimler
Tools: Python, ROS, MATLAB, CARLA, Rviz, Pyame, ProveTech-TA, Windows, Linux
University: Hochschule Ravensburg Weingarten (HRW)

- Developed MATLAB Graphical User Interface (GUI) for integrating Robot Operating System (ROS) and CARLA.
- Real world map Simulation in CARLA environment with GPS coordinates for testing.
- Various collision avoidance scenarios were generated from multiple sensor data.
- Path planning data was collected and trained to avoid collision with obstacles.

- Matlab GUI was made as a standalone application for both Linux and Windows platforms for Matlab and non-Matlab users.
- Sensor data from LIDAR, RADAR, CAMERA, IMU, GNSS were visualized in Rviz (ROS) and Pygame (Python).
- 3D environment of the autonomous vehicle was built from the information collected from LIDAR sensor.

EDUCATION

Master of Science **09/2017-08/2020**	**Hochschule Ravensburg-Weingarten (HRW), Weingarten** Field: Mechatronics

SKILLS

- React JS/ Node JS
- Microsoft Azure/ AWS
- Power BI
- Javascript/ HTML/ CSS/ PHP
- MongoDB/ PostgreSQL/ SQL Server
- Python/ C/ C++
- Englisch (Fluent) / German (B1)

- Version Control Systems(DevOps,GitLab, GitHub)
- Robot Operating System (ROS)
- MATLAB/ Simulink/ GNU Octave
- MxNet/GluonCV/ OpenCV
- PyQT/ Arduino/ Android Studio
- TensorFlow/ Keras

PROJECTS

Face and Smile Recognition **12/2018-02/2019**	**Tools**: Python, OpenCV, Webcam, TensorFlow/Keras - Haar Cascade was used to identify person's face and smile. - Input of user name and capturing their image using Webcam. - Separate ID's (labels) were created for different users. - Different user images were saved under their respective user ID's. - Number of images captured were varied and classified as three different cases. - The images were trained using both OpenCV and TensorFlow/Keras with the user ID's. - Once the images were trained the Webcam was initiated and corresponding users were identified from the trained data. - Based on the feedback, the accuracy was adjusted by varying different parameters such as number of images, light intensity, clarity of image, epos and so on.
Simulation of Autonomous Car (Scientific Project) **06/2018-11/2018**	**Tools**: Robot Operating System (ROS), Gazebo, Python, OpenCV - Simulation of a circuit/track environment with a Prius car in Gazebo. - The Camera sensor was attached to Prius and the images were published to the ROS nodes. - The sensor and camera information were subscribed from ROS nodes from respective topics. - The lane lines were sensed and extracted using their colors with OpenCV. - Desired twist and turn angles were scripted to avoid deviating from the sensed line. - Scripted the car to use the lane lines and the Sensor/Laser scan topics to keep circuiting the lane autonomously.

REFERENCES

- Olmatic GmbH