

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 : [Portfolio](#), [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 a scalable web application using a tech stack centered on Python, Node.js, and React.
- Engineered automated, conditionally triggered function apps in Python to construct robust data pipelines, adeptly managing the influx of data from both customers and sensor sources.
- These function apps were meticulously designed to efficiently process incoming data, facilitating seamless integration with SQL databases. Additionally, they were configured to serve this processed data through accessible endpoints, ensuring interoperability with various external REST APIs.
- Ensured secure authentication using Azure Key Vault secrets and B2C services, safeguarding user data and interactions.
- Designed and implemented creative and user-friendly UI/UX elements, enhancing the overall user experience.
- Engineered RESTful APIs, utilizing Python and Cosmos DB services for efficient data retrieval and processing.
- Capitalized on Azure cloud services, including Function App Services, Key Vaults, Cosmos DB, Storage Accounts, Azure B2C, and Service Principles, to construct serverless applications and automate workflows.
- Proficiently designed, optimized, and managed databases employing Cosmos DB, Stored Procedures, and SQL Server, ensuring efficient data storage and retrieval.
- Implemented DevOps practices, including CI/CD pipelines, to streamline development workflows and expedite deployments.
- Engaged in agile ceremonies, such as daily stand-ups, sprint planning, and retrospectives, to drive efficient project management and continuous improvement throughout the development lifecycle.

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)

- Designed a user-friendly MATLAB Graphical User Interface (GUI) to seamlessly integrate the Robot Operating System (ROS) with the CARLA simulator.
- Visualized sensor data obtained from LIDAR, RADAR, CAMERA, IMU, and GNSS in real-time, leveraging tools like Rviz in ROS and Pygame in Python.
- Generated and simulated various collision avoidance scenarios based on data from multiple sensors.
- Collected and processed path planning data to create intelligent collision avoidance strategies, ensuring safe navigation and obstacle avoidance.

EDUCATION

Master of Science

09/2017-08/2020

Hochschule Ravensburg-Weingarten (HRW), Weingarten

Field: Mechatronics

SKILLS

- React JS/ Node JS
- Python/ C++
- Microsoft Azure/ AWS
- Power BI
- Javascript/ HTML/ CSS/ PHP
- MongoDB/ PostgreSQL/ SQL Server
- 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

- Utilized Haar Cascade for robust detection of faces and smiles in real-time video feeds.
- Enabled user interaction by accepting user names and capturing their images via the webcam.
- Employed a structured labeling system to assign unique IDs to different users.
- Managed variable image quantities for each user, categorizing them into three distinct cases.
- Employed OpenCV and TensorFlow/Keras for image training with corresponding user IDs.
- Deployed the trained model to perform real-time recognition using the webcam.
- Fine-tuned accuracy by adjusting parameters like image quantity, lighting conditions, image clarity, and epochs based on feedback.

Simulation of Autonomous Car (Scientific Project)

06/2018-11/2018

Tools: Robot Operating System (ROS), Gazebo, Python, OpenCV

- Created a realistic circuit/track environment within the Gazebo simulation platform.
- Integrated a camera sensor onto a virtual Prius car and streamed images to ROS nodes.
- Subscribed to sensor and camera data from ROS nodes via designated topics.
- Utilized OpenCV for efficient detection and extraction of lane lines based on their color characteristics.
- Scripted the implementation of desired twist and turn angles to ensure the vehicle stays on track, avoiding lane deviations.
- Programmed the Prius car to autonomously navigate the circuit using the extracted lane lines and data from Sensor/Laser scan topics.

REFERENCES

- [Olmatic GmbH](#)