1 Introduction:

The project report 1 provides a comprehensive overview of the OdyssiX team's progress on integrating a Brain repository with a Raspberry Pi for an autonomous vehicle system as part of the Bosch Future Mobility Challenge 2025. Also, we have described the overall progress on the hardware integration with the software integration. Overall, the late arrival of the vehicles did influence our initial plans; however, the team has a valid schedule to recover the backlog, and meet all the minimum requirements for report 2.

2 Planned activities:

Samarth Prashant Bankar:

- Raspberry Pi configuration and setup.
- GitHub repository initialization and version control strategy.
- Installation and setup of project management tools (Jira).

Jash Parikh:

- Development of modular and scalable software architecture.
- Comprehensive project planning aligned with deliverables.

Aarsha Shah:

- In-depth documentation of the embedded platform.
- Research into libraries (e.g., Mbed, Ninja) and tools (e.g., CMake) critical for the Nucleo board.

3 Status of planned activities:

1. **Brain-Computer Integration with Raspberry Pi**
   Status: Ongoing 95%.

   Implementation:
   - Configured Raspberry Pi OS Lite and established an SSH connection.
   - Installed essential libraries and customized environment variables for compatibility with the brain interface.
   - Hosted the dashboard, but debugging is ongoing to enable vehicle control.

   Difficulties:

   - **Cooling System Issue**: Resolved by sourcing a missing fiber screw from local vendors.
   - **microSD Formatting**: Persistent issues were resolved by using `diskmgmt.msc` to delete partitions and reformat to FAT32.
   - **Code Integration**: Encountered path and environment configuration errors.

2. **Project management installation and setting up:**

   Status: 100% completed

   Implementation: Jira was configured for task allocation, deadline tracking, and backlog management.

   Difficulties: None

3. **GitHub repository** Initialization**:**

   Status: 100% completed

   Implementation: Established a version control strategy for collaborative development.

   Difficulties: None

4. **Software architecture:**

   Status: 100% completed

   Implementation: Created a preliminary architecture diagram detailing the data flow between sensors, actuators, and processing units. We have offered only a rough idea, of the architecture, but as we explore in depth the other hardware components we plan to improvise on the software part, and make it industry level optimized for the most optimal performance.

   Difficulties: None

5. **Project plan:**

   Status: 100% done

   Implementation: To make a thorough project plan for us as well as the mentor, so deadline of the tasks can be met time to time. The project plan is heavily influenced from the software architecture. We understand the proper synchronization of the software and the hardware is going make this project successful, so we do not want to compromise on the security of our network in anyway, and we definitely plan to emphasize our time and resources to optimize the network communication within the system.

   Difficulties: None

6. **Embedded platform documentation:**

   Status: Ongoing, 30% completed.

   Implementation: Set up the environment, flashed the Nucleo board, and initiated debugging with Putty.

   Difficulties: Limited documentation on libraries and tools such as Ninja and Mbed

4 General status of the project

We successfully flashed the microSD card with the brain code and started debugging for vehicle control. We optimized the Raspberry Pi configuration and explored advanced network communication setups for real-time data transfer. We also flashed the Nucleo board and configured its embedded environment to ensure compatibility with the Raspberry Pi. We identified a damaged camera belt and are exploring replacement options to restore functionality.

5 Upcoming activities

1. **Debugging Raspberry Pi Brain Code**: Finalize debugging to establish seamless communication and synchronization between the brain interface and the Raspberry Pi for effective control.
2. **Testing Track Development**: Construct a testing track with ramps, tunnels, and other elements as per competition protocols to validate the vehicle's performance in controlled environments.
3. **Nucleo Board Configuration**: Flash and configure the Nucleo board to enable real-time bidirectional communication with the Raspberry Pi.
4. **PiCam Integration for Object Detection**: Establish a data pipeline between the PiCam and Raspberry Pi to implement real-time object detection and obstacle avoidance using optimized computer vision algorithms.
5. **Virtual Testing Environment Deployment**: Install and configure a simulation environment to test and validate the system's end-to-end functionality under various scenarios.
6. **Input-to-Motor Signal Flow Implementation**: Develop and test the complete signal processing pipeline, converting camera input (visual data) into actionable motor control commands using a combination of computer vision, sensor fusion, and PID control algorithms.
7. **End-to-End System Validation**: Conduct extensive testing to ensure the integrated system, from camera input to motor output, operates reliably in both virtual and real-world environments. This includes real-time latency measurement, system optimization for performance, and fail-safe mechanisms for unexpected conditions.