**Bus Project**

|  |  |
| --- | --- |
| **First Name:\_\_\_\_\_\_Aastha\_\_\_\_\_\_**  **Last Name:\_\_\_Mistry\_\_\_\_\_\_\_\_\_\_**  **Student ID:\_\_\_\_N01550654\_\_\_\_\_\_\_\_\_** | **First Name:\_\_\_\_Harikant\_\_\_\_\_\_\_\_**  **Last Name:\_\_Sharma\_\_\_\_\_\_\_\_\_\_\_**  **Student ID:\_\_N0154713\_\_\_\_\_\_\_\_\_\_\_** |

**Project Overview:**

This report details the implementation and completion of the networking component for the TicTacToe game with AI using a Raspberry Pi and Sense HAT module. The project falls under the scope of the Communications Engineering course (IENG 2530) and is a part of the BUS Project. The primary objective of this component is to create an online scoreboard to store match scores or player moves in the cloud and optionally fetch and display these scores on the Raspberry Pi’s RGB LED matrix or a web page.

**Project Objectives:**

The project objectives outlined for the networking component are as follows:

1. Implement an online scoreboard to store match scores or player moves in the cloud.
2. Optionally fetch the scores from the cloud and display them on the Raspberry Pi’s RGB LED matrix or a web page.
3. Design the project considering various implementation possibilities, including networking methods, cloud service providers, and display options.
4. Set up the chosen cloud service (Google Cloud Platform recommended) and integrate it with the Raspberry Pi.
5. Develop Python code to send score or move data to the cloud database.
6. Install a web server (Apache or Flask) on the Raspberry Pi and create a webpage to display the game data stored on the cloud.
7. Utilize networking tools like Wireshark to measure network parameters between the Raspberry Pi and the cloud.
8. Enable the TicTacToe game to be played over the web using a Python web framework like Flask.

**Project Steps:**

1. **Design Phase:**
   * Consider various implementation possibilities, including batch update vs. live updates for the scoreboard, choice of networking (WiFi or wired Ethernet), and selection of cloud service provider (Google Cloud Platform recommended).
2. **Cloud Setup:**
   * Create a new spreadsheet on Google Sheets for storing scores or moves.
   * Set up Google API by creating a project, enabling API, adding credentials, and downloading the JSON key.
   * Share the Google Sheet with the generated email address and install the OAuth client if necessary.
3. **Raspberry Pi Setup:**
   * Install required software on the Raspberry Pi based on the design decision.
   * Integrate Python code to send score or move data to the Google Sheet in the cloud.
   * Test the functionality iteratively to ensure correct values are written to the Google Sheet.
4. **Web Server Setup:**
   * Install a web server (Apache or Flask) on the Raspberry Pi.
   * Create a webpage to display the game data stored on the cloud database.
   * Enable the TicTacToe game to be played over the web using Flask or similar Python web framework.
5. **Network Analysis:**
   * Utilize networking tools like Wireshark to measure network parameters such as HTTP/DNS/UDP packets, latency, packet loss, and bandwidth usage between the Raspberry Pi and the cloud.

**Stretched Goal:**

* Retrieve the score data from the cloud and display it on either the Raspberry Pi’s RGB LED matrix or a web page.
* The implementation can be decided based on the project team's discretion and creativity.

**Project Completion:**

The project was successfully completed with the following deliverables:

* Python scripts (app.py, main.py, ttsheet.json) for integrating the TicTacToe game with Google Sheets API and implementing the online scoreboard.
* Setup instructions for cloud integration and Raspberry Pi configuration.
* Implementation of a web server to display game data on a webpage.
* Utilization of networking tools for measuring network parameters.
* Achievement of the stretched goal by enabling the display of score data retrieved from the cloud.

**Conclusion:**

The networking component of the TicTacToe game with AI using Raspberry Pi and Sense HAT module has been effectively implemented, meeting the project objectives and requirements. The integration with Google Sheets API provides a robust cloud-based solution for storing and retrieving game data. Additionally, the web server setup enables convenient access to game information over the web. The project demonstrates successful collaboration between hardware (Raspberry Pi), software (Python), cloud services (Google Cloud Platform), and networking technologies, showcasing practical applications of communications engineering principles.  
  
**Photos:**A screenshot of a computer

Description automatically generatedA hand holding a small square device with a green light

Description automatically generated