SMART PARKING

LOADING AND PREPROCESSING THE DATASET

TRAINING

VALIDATION

SUBSET

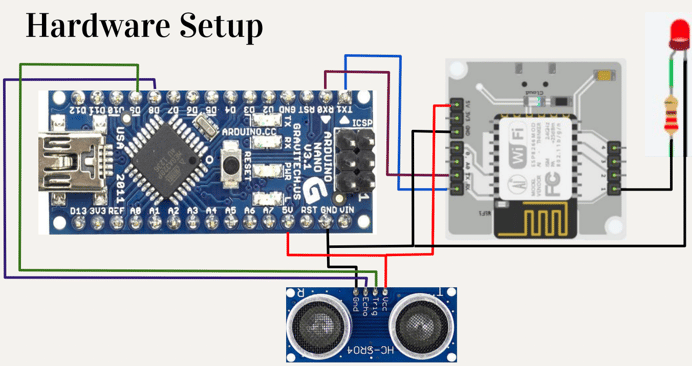
## Things used in this project:

**Hardware Components:**

|  |  |
| --- | --- |
| [Bolt Wifi Module](https://shop.boltiot.com/pages/iotmltraining) | X 1 |
| [Arduino Nano](https://www.amazon.in/dp/B077TJ8FCF/ref=cm_sw_r_apan_i_5R093BB7AXT2M5VJJRB9?_encoding=UTF8&psc=1) | X 1 |
| [Ultrasonic Sensor](https://www.amazon.in/dp/B01I1ZTPJC/ref=cm_sw_r_apan_i_R4JVNNNMRN8TFEQYDKKV?_encoding=UTF8&psc=1) | X1 |
| [Jumper Wires](https://www.amazon.in/dp/B074J9CPV3/ref=cm_sw_r_apan_i_R4P9Z649YHDFDA65XNH4?_encoding=UTF8&psc=1) | x10(6 male-male,  6 male-female) |
| LED(included with Bolt IoT training kit) | x1 |
| Resistor(10 kΩ) | x1 |
| Bread Board | x1 |

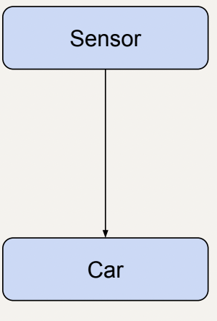
**Software and online services:**

|  |
| --- |
| [Bolt Cloud](https://cloud.boltiot.com/) |
| Python IDE |
| Arduino IDE |
| SQLite |
| Vue JS |
| Flask |

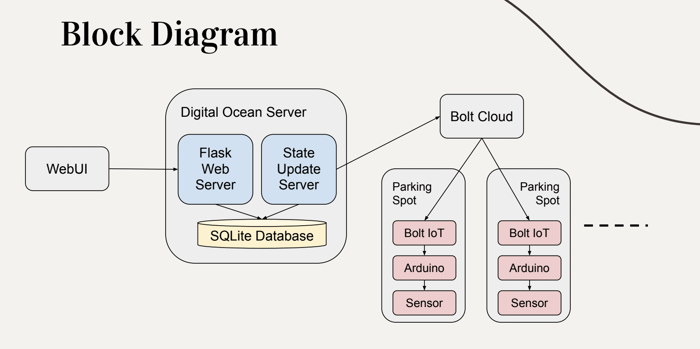
**Hardware Setup:**

**Software Programming:**

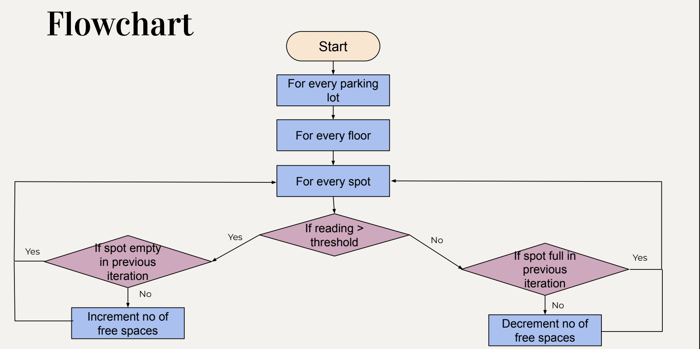
**Basic Working:**

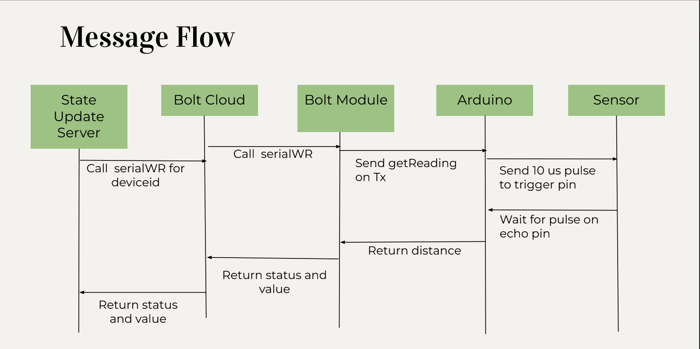
● Detecting empty parking spot  
● Indicating empty parking spot  
● Display total number of empty  
parking spots

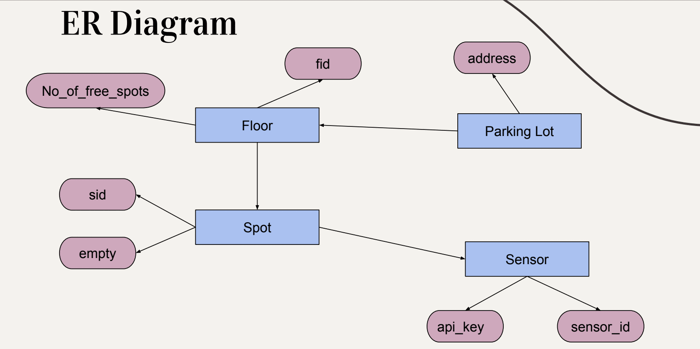
**Block Diagram:**



**Flowchart:**

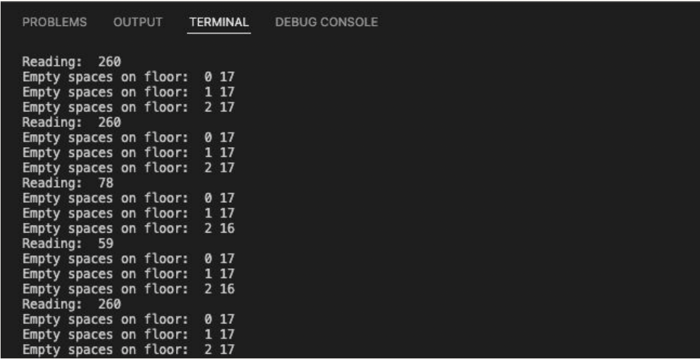
**Message Flow:**

**ER Diagram:**

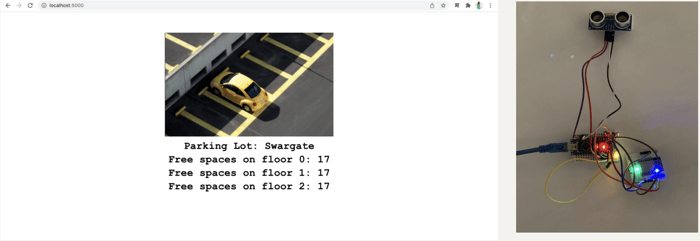


**OUTPUT:**

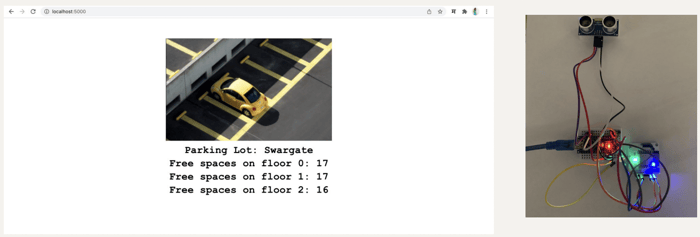
**Debug Log Output:**

****

**Output - Spot is empty:**

****

**Output - Spot is full**

****

Below is a video demonstration:

**Conclusion:**

● Automated IoT based system  
● Number of free spaces displayed  
● Glowing LED to indicate presence of empty spot  
● Database updated every 10 seconds  
● Manual assistance eliminated  
● Time wastage eliminated

SOFTWARE REQUIREMENETS 1.

Python Python is an interpreted high for programming. Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and programming. It is used in Web and Internet Development, Software Development and Business Applications; it has many more applications as it has features like Multiple Programming Paradigms, Compatible with Major Platforms and Systems and Robust Standard Library.

Hypertext Markup Language (HTML) is the underlying markup language of the World Wide Web [6].HTML is a markup language that web browsers use to interpret and compose text, images, and other material into visual or audible web pages. The web app is designed using HTML.Hypertext Markup Language (HTML) is the standard markup language for creating web pages and applications. Web receives HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically.

3. Django

Django is a source web, written in Python, which follows the model-view-template (MVT) architectural pattern. It consists of an object-relational mapper (ORM) that mediates between data models (defined as Python classes) and a relational database ("Model"), a system for processing HTTP requests with a web templating system ("View"), and a regular-expression-based URL dispatcher ("Controller").Some of its features are a lightweight and standalone web server for development and testing, a form serialization and validation system that can translate between HTML forms and values suitable for storage in the database, and a caching framework that can use any of several cache methods. In this project we have used django version 2.1, which is the latest version released by Django foundation.

HARDWARE REQUIREMENTS

1. Raspberry Pi

The Raspberry Pi is a series of small single-board computers developed by the Raspberry Pi Foundation. The first-generation Raspberry Pi 1 Model B was released in February 2012. Raspberry Pi 3 Model B was released in February 2016 with a 1.2 GHz 64-bit quad core processor, on-board Wi-Fi, Bluetooth and USB boot capabilities. Raspbian is a free operating system based on Debian GNU/Linux and optimized for the Raspberry Pi hardware. Here we are using Raspberry Pi 3B+.It has 40 GPIO pins

1. IR sensors

IR sensors are used to detect the presence of a vehicle. The radiations are not visible to naked eye IR sensors detect the heat and motion of an object or person. Thermal radiations are detected by passive IR sensors. The resistances and the output voltages, change in proportion to the magnitude of the IR light received [1]. TheIR transmitter sends an infrared signal that, in the presence of a reflecting surface (especially if it is white), "bounces" in various directions, including the one along which the radiation hits the IR receiver, which captures the signal by detecting the object, and signaling it through one of its pins.

1. Jumper Wires

A jump wire is an electrical wire, or group of them in a cable, with a connector or pin at each end, which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering. Jumper wires are wires that have connector pins at each end, allowing them to be used to connect two points to each other without soldering. Jumper wires are typically used with breadboards and other prototyping tools in order to make it easy to change a circuit as needed [2]

1. Perforated PCB

Perfboard is a material for prototyping electronic (also called DOT PCB). It is a thin, rigid sheet with holes predrilled at standard intervals across a grid, usually a square grid of 0.1 inches (2.54 mm) spacing. These holes are ringed by round or square copper pads, though bare boards are also available. Inexpensive perfboard may have pads on only one side of the board, while better quality perfboard can have pads on both sides (plate-through holes). Since each pad is electrically isolated, the builder makes all connections with either wire wrap or miniature point to point wiring techniques. Discrete components are soldered to the prototype board such as resistors, capacitors, and integrated circuits. The substrate is typically made of paper laminated with phenolic resin (such as FR-2) or a fiberglass-reinforced epoxy laminate

1. IMPLEMENTATION OF THE SYSTEM

.A new project is started in django in the Raspberry Pi and a new app is created, then appropriate changes are made in the framework. 2. The code for the main logic is written in Python, and the code for web app is written in HTML and are saved in the project. 3.IR sensors detect whether the parking space is occupied by a vehicle or if the space is empty. The data is the send for further processing to Raspberry Pi. 4. The monitored data is analyzed in real time and the optimum control information is determined and transmitted to django framework. 5. When a user clicks on the URLs, django goes to the URL file and decides which function to fire in views and gives the end result in form of html template response in the browser app that gives the status of the parking spot.

1. RESULT AND DISCUSSION

This designed automatic smart parking system which is simple, economic and provides effective solution to reduce carbon footprints in the atmosphere. It is well managed to access and map the status of parking slots from any remote location through web app. Thus, it reduces the time of finding the parking slots in any parking area and also it eliminates unnecessary travelling of vehicles across the filled parking slots in a city. So, it reduces time and it is cost effective alsO