Table of Contents

1. Introduction

Home Application

"Connected fridge-freezer with various sensors such as temperature, and capability to auto-order items"

Multimedia-capable intelligent appliances have been making their way into our daily lives. The rapid advancement of computer technology and the widespread usage of the Internet have made the smart home one of the most well-known applications for intelligent appliances. One area where such sophisticated appliances have been used is the kitchen. Given that the consumer must buy the entire refrigerator, the existing products are pricey. Any existing refrigerator can be transformed into an intelligent, affordable appliance utilising sensor using the Smart Refrigerator module. The recipe is displayed on LCD in a previously set method based on the ingredients that are now in the refrigerator. Our refrigerator has the ability to sense and monitor its contents. It can also remotely alert the user via an Android application when certain products are running low. By including a link to the internet retailer of that specific item, it also makes it easier to buy rare products (Velasco et al., 2020).

The Internet of Things (IOT) concept necessitates ubiquitous connectivity to billions of disparate devices. Recent years have seen a tremendous increase in IOT devices in the context of smart homes, which has envisioned a wide range of unique services and uses. A smart home's kitchen is one of the most crucial rooms since it includes numerous appliances that improve household services. Our project is centred on the smart refrigerator. There have been numerous attempts to construct the smart refrigerator, but none of them have proven financially or energy-efficient. The user is unable to maintain track of the food items in the refrigerator due to modern living and the fast-paced surroundings (Mahajan et al., 2017).

Despite the industry's efforts to create a smart refrigerator, the present or existing technology is still not economical or energy-efficient. For a typical household user who is unfamiliar with the intricate workings of the smart refrigerator, the technology is either too sophisticated or confusing. Most locations still have inadequate internet connectivity, and there is little network connectivity, either in the form of slow internet speeds or bad assistance. The crucial information about the goods, such as the expiration date, is not uniformly recorded by the barcode. The networked home or smart home environment lacks sufficient security to prevent data leaks from the house. Attackers may jeopardise the user's and the home's privacy. The smart system can be controlled by remote devices using any operating system. Since there is no industry standard, different companies produce goods that adhere to different standards. The internet refrigerator, often known as a smart refrigerator, is used to track the contents and provide alerts when certain foods become scarce. When energy conservation is possible, offer ON/OFF control through cell phone, etc. The concept of internet-connected home appliances or the smart home environment has long been considered to be the future and the next big thing (Fujiwara et al., 2018).

The Internet of Things (IoT) is a global network of interconnected computing devices, mechanical and digital machinery that may exchange data over a network without requiring human-to-human or human-to-computer contact. A collection of specialised transducers with a communications network for monitoring and recording conditions at various locations makes up a wireless sensor network. The creation of smart refrigerators has been the subject of prior investigations. The Intelligent Refrigerator made use of infrared (IR) emitters to identify the products and alert the user via SMS and the shop owner via Ethernet network of the state of the products inside the refrigerator. The Smart Refrigerator measures the weight of the items within the refrigerator using a weight sensor or load cell. Through Bluetooth, it alerts the user to its status. The Smart Fridge employed a photodiode and a Node MCU, respectively, as the sensor to detect the product and the microcontroller to handle the sensor's data. A refrigerator with an overhead camera that recognises faces to provide users access and utilises RFID to find and track the contents is used to store medical supplies online. The Low-Cost Smart Refrigerator, on the other hand, utilised a camera to take images of the food within the refrigerator, an IR distance sensor, and a light sensor to determine if the refrigerator was open or closed (Mani et al., 2020).

The smart refrigerator uses image processing to recognise the inventories and things inside the appliance, however because the recognition was carried out using template matching, the image processing component crashed. Next, the High Chest smart freezer was created to support energy conservation and appropriate food storage. A system for managing products was suggested to keep track of perishable and non-perishable goods using GSM and Android applications (Saidur, Masjuki and Choudhury, 2002).

A customised Coke machine at Carnegie Mellon University served as the first internet-connected appliance, reporting its inventory and the temperature of freshly loaded drinks. The idea of a network of smart devices was initially suggested in 1982. Through the Auto-ID Center at MIT and related market study publications, the idea of the internet of things first gained popularity in 1999. Kevin Ashton, one of the original Auto-ID Canter founders, believed that radio-frequency identification (RFID) was necessary before the internet of things could exist. If every person and thing in daily life had an identification, then computers could manage and keep track of them. In addition to using RFID, near field communication technology can be used to tag objects (Suhuai Luo, Jin and Li, 2009).

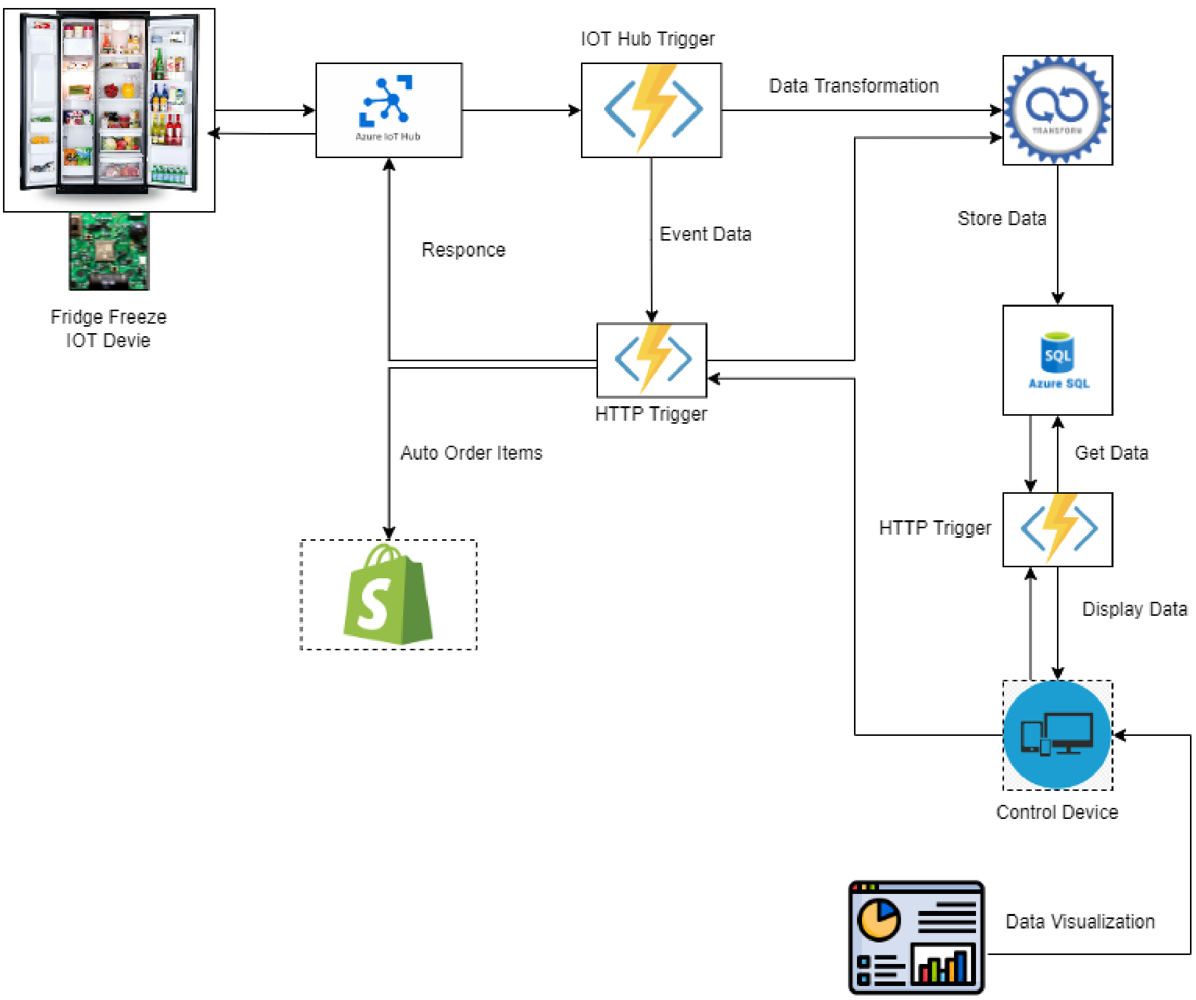
The concept of tying home appliances to the internet (Internet of Things) had gained popularity and was regarded as the upcoming big thing by the late 1990s and early 2000s. The Internet Digital DIOS, the first internet-connected refrigerator, was introduced by LG in June 2000. A refrigerator that has been designed to detect the types of goods being stored inside it and keep track of the stock by barcode or RFID scanning is referred to as a "Internet refrigerator" (also known as a "Smart refrigerator"). This type of refrigerator frequently has the capability to recognise when a food item requires replenishment. Given that it cost more than $20,000 and was perceived by consumers as being unneeded, as well as the fact that the problems it claimed to answer were vague, this refrigerator was a failure. For instance, many juice bottles are transparent, serving as a visual cue that a purchase will eventually be necessary. Similarly, vegetable drawers are transparent and frequently contain items that have been removed from packages, eliminating the need for bar codes in inventory, which required manually entering descriptions and dates. Additionally, it is not often possible to solve the use case of the gadget being able to notify users of forthcoming purchases when there are frequently numerous buyers in a household that interact informally (James, Onarinde and James, 2016).

By 2013, the Internet of Things' original concept had changed as a result of the convergence of several technologies, including wireless communication, the Internet, embedded systems, and microelectromechanical systems (MEMS). As a result, the conventional domains of automation (including home and building automation), wireless sensor networks, control systems, embedded systems, and others all contribute to enabling the internet of things (IoT). The Internet of Things (IoT) is a network of interconnected computing devices, mechanical and digital machinery and people that may exchange data over a network without requiring human-to-human or human-to-computer contact. A collection of specialised transducers with a communications network for monitoring and recording conditions at various locations makes up a wireless sensor network (Patel, Patel and Scholar, 2016).

The study's overarching goal is to build a sensor network-based IoT-based refrigerator inventory monitoring system that will inform the user of the contents of the refrigerator.

2. Cloud Architecture Flow Diagram

The architecture diagram of the IOT device used in this application is:



Flows of Application

- Fridge Freezer IOT Device to IOT Hub.
- IOT Hub Trigger Azure Function to Azure SQL Database
- HTTP Trigger Azure Function from SQL Database to Web App
- IOT Hub Trigger to HTTP Triger Event call
- Conditional Auto Order Flow
- Control App Device to IOT Hub

2.1 Fridge Freezer IOT Device to IOT Hub

In this flow, the sensor device circuit (IOT Device) that is installed in the fridge freezer will generate the data (Temperature, Items Details etc.), and then generated data sent to Azure IOT Hub.

2.2 IOT Hub Trigger Azure Function to Azure SQL DB

"IOT Hub Trigger Azure Function" is a type of Azure Function that will trigger when Azure IOT Hub receive any data from Fridge Freezer IOT Device. We will transform the data into a specific schema and pass the data to SQL API that will save the data to the Azure SQL Database.

2.3 HTTP Trigger Azure Function from SQL DB to Web App

"HTTP Trigger Azure unction" is a type of serverless function that will trigger when HTTP request will be called. It will get data from the Azure SQL Database using SQL API. Then we will call this "HTTP Trigger Azure Function" in our Frontend Framework (React) that will visualize the data in graphs and charts etc.

2.4 IOT Hub Trigger to HTTP Event Call

When a specific event occurs like temperature increase a given limit an automatic event occurs like low the temperate or shut down the fridge etc. will occurs and the event data will pass from the IOT Hub Trigger Azure Function that will update the event and call the Azure IOT Hub and update the Fridge state.

2.5 Conditional Auto Order Flow

This flow will call when an item quantity will low then the given quantity and it will automatically place order on online shopify sites. We will call APIs to place order.

2.6 Control App Device to IOT Hub

A web app where we visualize data can also trigger IOT Hub Device by just clicking of buttons and after pressing buttons like shut down fridge, the event data will pass from Web App to IOT Hub through HTTP Trigger Azure Function that will update the state.

3. Cloud Services

The Microsoft Azure Clous Services that we will use in the implementation of this Fridge Freezer IOT Device Application are as follows:

- Azure IOT Hub
- Azure SQL Database
- Azure Function
- Azure Web App

Explanation

3.1 Azure IOT Hub

Azure IoT Hub is Microsoft's IoT connector to the cloud. It is a fully managed cloud service that enables reliable and secure two-way communication between millions of IoT devices and back-end solutions. Azure IoT Hub enables IoT response by providing easy, reliable communication between billions of IoT devices and a response backend hosted in the cloud. IoT Hub supports multiple messaging styles consisting of telemetry from tools to the cloud, file uploads from devices, and request/response strategies for controlling devices from the cloud. IoT Hub video display unit and tracking events consisting of tool appearance, device failure, and tool connection (Stackowiak, 2019).

Features

1. Connect bi-directionally to billions of IoT devices.
2. Validate each device in advanced IoT solutions.
3. Use IoT Hub Device Provisioning Service to register devices for scaling.
4. Use device management to Managing IoT Devices at Scale.
5. Cloud - Peripherals to Extend Functionality.

Uses

We will use Azure IOT Hub to connect out Fridge Freezer Sensor Device to the Azure IOT. To connect any sensor device to the internet we will use a central device (Hub) that will communicate bi-directional from sensor device to IOT Hub and IOT Hub to sensor device that acts in a real time environment. IOT hub has different events behavior that will trigger when a specific condition occurs. In our application different events will be triggered like High Temperature, Low Temperature, Shut down Fridge and Auto Order Inventory bases on different conditional behavior (Klein, 2017).

3.2 Azure SQL Database

Microsoft Azure SQL Database is a globally distributes and fully managed Platform as a Service (PaaS) database engine that handles most of the administrative functions provided by the database, including backups, patches, upgrades, and monitoring, with minimal user interaction.

Azure SQL Database requires a patched operating system and a stable version of SQL Server. It also helps to create a highly availablility and performant data storage layer for your applications (Reagan, 2017).

Azure SQL Database takes full advantage of the cloud platform. With a goal of 99.99% availability, we are committed to virtually zero downtime for the databases we host. High-availability architecture is used to protect data from failures and frees users from worrying about maintenance or potential failures. This architecture is also resilient, allowing organizations to build robust apps with retry logic to prevent loss of functionality or connectivity in the event of a failure (Ward, 2020).

Features

The following are the features of the Azure SQL Database (De Tender, 2019).

Scalability: Azure SQL Database is scalable. We can use single database and Elastic pools, multiple instances of SQL database to scale up and we can also use availability zone to scale it up.

Performance: Azure provides a 99.9999% performance factor, which means that there will be a very low-down time. The SQL Database services will provide us a 100% up time that will increase our application performance.

Security: In Azure, the databases are encrypted by default. It uses different keys to maintain the security of the database.

Usage

We are using Azure SQL Database to store the sensor data into the database for future use. Different Events data also used to save in the SQL database in a predefined schema. First sensor data will come to IOT Hub Trigger Function and then data will pass to the Data Transformation Method that will transform the data into a specific type and then we will save the data. The saved data can be used in the future predictions and Analysis.

Sample Data Template

The Conceptual data model that I am assuming is as follows:

```
{
    SensorName: String,
    Temperature: Int,
    EventName:String,
    Items:List
    DateTime:DateTime
}
```

3.3 Azure Functions

Azure Functions is a serverless concept with a cloud-native design that allows you to deploy and run code without lacking server infrastructure, web servers, or another configuration. We can write Azure Functions in multiple languages, including JavaScript, C#, Java, TypeScript, and Python (Kurniawan and Lau, 2019).

- Azure Functions are scalable. As execution demand increases, more resources are automatically allocated to the provider, and as demand decreases, all large resources and application instances are automatically reduced.
- Azure skills are of acceptable quality as small apps may run independently from various websites. Common Azure functions include sending emails, initiating backups, processing orders, project planning including database clean-up, sending notifications, messaging, and processing IoT statistics.

Usage

We are using two types of Azure Functions in this Conceptual IOT Application.

- Azure IOT Hub Trigger Function
- Azure HTTP Trigger Function

3.3.1 Azure IOT Hub Trigger

Azure IOT Hub Trigger is a type of Azure function that will trigger when IOT Hub that is connected to the IOT Hub Trigger Azure Function will receive any data from the fridge freezer sensor device. We will pass the API keys in this function to connect it to the IOT Hub. Whenever IOT Hub will receive any type of data that is defined here and it will automatically call this function that will receive the stream data in a Realtime environment.

3.3.2 Azure HTTP Trigger

This is also a type of Azure Serverless Function. This Azure Function will trigger when an HTTP request is called. We are using this function in different parts of application.

First, we are using this function to get the data from the SQL Database. The data that is being generated from the Sensor device will pass to the SQL Database from the IOT Hub and IOT Hub Trigger Functions. We will call this function in our web app that will show the data onto the webpage

Secondly, we are using this Azure Function in the middleware to act as different types of events. To call auto response events we will use this Azure Function that will automatically trigger based of different logical conditions that are based on temperature and Items quantity. We also use this function to get the user inputs to control the fridge freezer device.

3.4 Azure App Service

Azure App Carrier uses our own cloud services to rapidly build apps, quickly and easily build organization-centric website and mobile applications for any platform or tool, and make them scalable, reliable and save. It helps you deploy and install on high-performance cloud infrastructures.

Azure Web Apps is a cloud computing-based website hosting platform built and operated by Microsoft. It's a platform-as-a-service that runs in multiple frameworks and lets you publish web apps written in a variety of programming languages, including Microsoft's own and third-party languages.

Usage

We are using Azure App Service as a website to show the data on the web page. Different data visualization techniques will be used to show up the data in different formats and in different charts etc. We also call different events to control this conceptual IOT Application. For Example, if we saw on website that temperature is low then a specific range then low up the Fridge speed etc. We just press on a Low Temperature button and it will automatically call HTTP Trigger function with event type and data and the called Azure Function will send event data to the IOT Device and a specific event occurs in a real-time behavior.

4. IOT Application Security Considerations

The security considerations that we will use to increase the security of this Conceptual IOT Application are as follow:

1. Software Testing and Updating
2. Avoid Universal Plug & Play Features
3. Data Encryption
4. Secure Networks
5. User Awareness Regarding IOT Security

Explanation

4.1 Software Testing and Updating

Testing is a process in which we evaluate and verify that a software product or application performs its intended functions. Benefits of testing include avoiding bugs, reducing development costs, and improving performance. So, we will test our complete application on different parameters to increase the application security.

Proper updating of software’s is also very important to increase the application security factor. Latest version of software’s can increase application security at a very higher level. Always use updated software because it can have some high-level current market security techniques which can help you to increase the application security at a very higher level (Kassab, DeFranco and Laplante, 2017).

4.2 Avoid Universal Plug & Play Features

Some developers enable auto plug and play features that are used to interact with other services to use in your own application, this can breach security and an unauthentic website or server can misuse your data which can cause security breach. So, we have to first investigate the Third-party services whether they save our data securely or not. Azure also provides a security of data that ensures that your data is very secure with highly availability as data has three copies in a data center and all over the network (Miller et al., 2001). So, if you using Microsoft Azure cloud then data security is auto maintained by the Azure.

After proper investigating the third-party services, you can embed it to your application and use their services, but again be aware to enable the auto plug and play feature.

4.3 Data Encryption

IoT security is about protecting the internet connected devices from threats and compromises by protecting, identifying, and monitoring risks while remediating vulnerabilities in various devices that can pose security risks to an organization. Data should be encrypted before it can send over the network. Data should be encrypted so that only the Application owner can view the data. We can use different keys to encrypt our data like public and private keys. Private key is only owned by the data owner and he/she can only view their data. No any person can view the data which can increase the data security (Kefa Rabah, 2005).

4.4 Secure Networks

With data security we also have to secure the network on which data is traverse. Network security is a set of tools and technologies designed to protect the usability and integrity of an organization's infrastructure by preventing various threats from entering or propagating within the whole network (Chandra et al., 2015).

We can use HTTPS protocols and REST APIs to send and receive data over the network. A Role Based Access Control (RBAC) technique is uses to secure the network security. Firewalls also be used to allow and block specific users. Whitelisting and blacklisting the user based on their network consumption to increase the network security.

4.5 User Awareness Regarding IOT Security

Since the Internet of Things is a very new technology, users still have to get used to its quirks and peculiarities.

- People have largely learned their own security when it comes to computer phishing, virus and malware attacks, and Internet fraud. Peoples should have enough knowledge to secure their WIFI networks and how to secure the credit cards online.
- But when we deal with the IoT security issues, lack of education and user neglect can also be to blame, as much as it is on the shoulders of manufacturers.
- The user lack of awareness is probably the biggest security issue for the use of Internet of Things.

The user must be aware about Social Engineering, because attackers can reveal the user secrets by just talking to the target users. So, we have aware the IOT application user so that they can follow best practices to increate this Conceptual IOT application.

5. DevOps Testing Strategies

The DevOps testing strategies that can be used for the testing of this conceptual IOT application are as follows:

1. Compatibility Testing
2. Reliability and Scalability Testing
3. Load Testing
4. Functional Testing
5. Performance Testing

Explanation

5.1 Compatibility Testing

Compatibility testing is a testing technique that is used to test the application in the same environment but having different versions (Yoon et al., 2008). We will test the conceptual IOT application on different versions of the same software to check whether it outputs the same performance and accuracy or not. We will check both the application-to-application level and hardware to hardware level compatibility to get the desired accuracy.

5.2 Reliability & Scalability Testing

Reliability and scalability are key to building IoT test environments that simulate sensors using virtualization tools and technologies. Reliability testing is performed by team members to ensure that the application behaves and functions consistently under all different environmental conditions and over time. This ensures that the product is error-free and reliable for its intended purpose (Iyer, Gupta and Johri, 2005). We will test our conceptual IOT Application on different parameters to check whether it performs well or not under different circumstances.

5.3 Load Testing

Load testing is a type of software testing in which test the whole application as all (Jiang and Hassan, 2015). We will apply different loads to check whether is respond well under different load conditions or not. We will use the Sensor IOT device all over the day on different load to check whether it performs good under different load factors.

5.4 Functional Testing

Functional testing is a testing technique in which the individual parts of the conceptual IOT application are tested to check whether they are performing well (Tomić and Vlajić, 2008). We will test individual parts in terms of their speed, performance and security factors. If we found some problems with the functional parts, then we can eliminate the thread at the functional level that can increase the application performance.

5.5 Performance Testing

Performance testing is a testing technique used to determine how does a system respond in terms of performance, stability and availability under a given condition (Kunhua Zhu, Junhui Fu and Yancui Li, 2010). We will check the performance of our conceptual IOT application to check whether it performs well in terms of speed and stability. We will evaluate its speed factor under different conditions to make its performance better.

6. References

Chandra, P., Rama, S., Behera, P., Purna, C., Sethi, Kumar, P. and Reader, B. (2015). Methods of Network Security and Improving the Quality of Service -A Survey Network traffic analysis for a software defined network using bandwidth on demand View project CSI2020 (Spinger Conference) ‘Digital Democracy -IT for Change’ View project Methods of Network Security and Improving the Quality of Service -A Survey. 5(7).

De Tender, P. (2019). Azure SQL Database Deployment using the Azure Portal. *Introducing Azure SQL Database*. doi:10.1007/978-1-4842-5276-5_2.

Fujiwara, M., Moriya, K., Sasaki, W., Fujimoto, M., Arakawa, Y. and Yasumoto, K. (2018). A Smart Fridge for Efficient Foodstuff Management with Weight Sensor and Voice Interface. *Proceedings of the 47th International Conference on Parallel Processing Companion*. doi:10.1145/3229710.3229727.

Iyer, L.S., Gupta, B. and Johri, N. (2005). Performance, scalability and reliability issues in web applications. *Industrial Management & Data Systems*, 105(5), pp.561–576. doi:10.1108/02635570510599959.

James, C., Onarinde, B.A. and James, S.J. (2016). The Use and Performance of Household Refrigerators: A Review. *Comprehensive Reviews in Food Science and Food Safety*, 16(1), pp.160–179. doi:10.1111/1541-4337.12242.

Jiang, Z.M. and Hassan, A.E. (2015). A Survey on Load Testing of Large-Scale Software Systems. *IEEE Transactions on Software Engineering*, 41(11), pp.1091–1118. doi:10.1109/tse.2015.2445340.

Kassab, M., DeFranco, J.F. and Laplante, P.A. (2017). Software Testing: The State of the Practice. *IEEE Software*, 34(5), pp.46–52. doi:10.1109/ms.2017.3571582.

Kefa Rabah (2005). Theory and Implementation of Data Encryption Standard: A Review. *Information Technology Journal*, 4(4), pp.307–325. doi:10.3923/itj.2005.307.325.

Klein, S. (2017). Ingesting Data with Azure IoT Hub. *IoT Solutions in Microsoft's Azure IoT Suite*, pp.57–70. doi:10.1007/978-1-4842-2143-3_4.

Kunhua Zhu, Junhui Fu and Yancui Li (2010). Research the performance testing and performance improvement strategy in web application. *2010 2nd International Conference on Education Technology and Computer*. doi:10.1109/icetc.2010.5529374.

Kurniawan, A. and Lau, W. (2019). Introduction to Azure Functions. *Practical Azure Functions*, pp.1–21. doi:10.1007/978-1-4842-5067-9_1.

Mahajan, M., Nikam, R., Patil, V., Dond, R. and Prof, A. (2017). Smart Refrigerator Using IOT. *International Journal of Latest Engineering Research and Applications (IJLERA)*, [online] 02(03), pp.86–91. Available at: http://www.ijlera.com/papers/v2-i3/part-II/31.201703127.pdf.

Mani, P., Santhoshkumar, A., Kumar, K.M., Siva, R., Gnanamani, S., Janarthanam, H. and Ponnappan, V.S. (2020). Design of smart container in refrigerator for convenience usage. *3RD INTERNATIONAL CONFERENCE ON FRONTIERS IN AUTOMOBILE AND MECHANICAL ENGINEERING (FAME 2020)*. doi:10.1063/5.0034288.

Miller, B.A., Nixon, T., Tai, C. and Wood, M.D. (2001). Home networking with Universal Plug and Play. *IEEE Communications Magazine*, 39(12), pp.104–109. doi:10.1109/35.968819.

Patel, K., Patel, S. and Scholar, P. (2016). Internet of Things-IOT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges. [online] doi:10.4010/2016.1482.

Reagan, R. (2017). Azure SQL Databases. *Web Applications on Azure*, pp.77–137. doi:10.1007/978-1-4842-2976-7_4.

Saidur, R., Masjuki, H.H. and Choudhury, I.A. (2002). Role of ambient temperature, door opening, thermostat setting position and their combined effect on refrigerator-freezer energy consumption. *Energy Conversion and Management*, 43(6), pp.845–854. doi:10.1016/s0196-8904(01)00069-3.

Stackowiak, R. (2019). Azure IoT Hub. *Azure Internet of Things Revealed*, pp.73–85. doi:10.1007/978-1-4842-5470-7_4.

Suhuai Luo, Jin, J.S. and Li, J. (2009). *A smart fridge with an ability to enhance health and enable better nutrition*. [online] ResearchGate. Available at: https://www.researchgate.net/publication/228359020_A_smart_fridge_with_an_ability_to_enhance_health_and_enable_better_nutrition.

Tomić, B. and Vlajić, S. (2008). Functional testing for students. *ACM SIGCSE Bulletin*, 40(4), pp.58–62. doi:10.1145/1473195.1473221.

Velasco, J., Alberto, L., Ambatali, H.D., Canilang, M., Daria, V., Liwanag, J.B. and Madrigal, G.A. (2020). Internet of things-based (IoT) inventory monitoring refrigerator using arduino sensor network. *Indonesian Journal of Electrical Engineering and Computer Science*, 18(1), p.508. doi:10.11591/ijeecs.v18.i1.pp508-515.

Ward, B. (2020). What Is Azure SQL? *Azure SQL Revealed*, pp.39–80. doi:10.1007/978-1-4842-5931-3_2.

Yoon, I.-C., Sussman, A., Memon, A. and Porter, A. (2008). Effective and scalable software compatibility testing. *Proceedings of the 2008 international symposium on Software testing and analysis - ISSTA '08*. doi:10.1145/1390630.1390640.