



**INTERNATIONAL JOURNAL OF RESEARCH AND  
ANALYTICAL REVIEWS (IJRAR) | IJRAR.ORG**  
An International Open Access, Peer-reviewed, Refereed Journal



**FRANCIS XAVIER  
ENGINEERING COLLEGE**  
**AN AUTONOMOUS INSTITUTION**

ACCREDITED BY NBA  
ISO 9001:2015 Certified | DST-FIST Supported Institution  
Recognized under Section 2(f) & 12(B) of the UGC Act, 1956



PROGRESS THROUGH KNOWLEDGE

# **Integrated Foolproof Vending System**

*Submitted by*

**LOGAPURUSHOTHAMAN T**

**Reg. No. 95072157008**

*in partial fulfillment for the award of the degree of*

**MASTER OF TECHNOLOGY**

**IN INFORMATION**

**TECHNOLOGY**

**FRANCIS XAVIER ENGINEERING COLLEGE**

**(Autonomous)**

**TIRUNELVELI – 627 003**

## ABSTRACT

Automated Vending Technology has become legacy now. Digitized version of Automated Vending Technology is termed as Autonomous Vending Technology. Intelligence added to has made it an iAutonomous Vending Technology. Using this Technology, Internet based retailing has become possible. Human intervention needed only at refilling part. Other than that, all the associated operations are taken care by the machine itself. It helps every common man to start or involve in retail sales based business at any moment. Biggest challenge in doing a retail business in accountability of stock, periodic monitoring of product availability, failure handling, managing supply chain, etc... But now, we have made all this operations into a robotic operations. Also, this helps to dispense products with any credentials.

## CHAPTER I

## INTRODUCTION

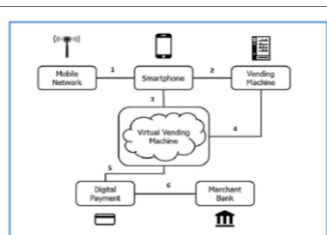


Figure 1. Components and users of a vending machine

### 1.1 Managed Cloud services

Managed cloud services are the partial or complete management and control of a client's cloud platform, including migration, maintenance and optimization. By using a managed cloud service provider, a business can ensure its cloud resources run efficiently. Outsourcing cloud management also allows businesses to avoid new hiring and training costs.

Managed cloud services can provide private, public and hybrid cloud environments. Working with a managed cloud services provider is a collaborative process. To determine the best cloud resources for the user's IT infrastructure, the managed cloud services vendor evaluates the user's applications and data. Then, they'll create a custom plan detailing the steps to build and run an optimized cloud environment.

Users adopt managed cloud services throughout the cloud solution lifecycle. These services can be used to help with initial adoption or can be provided continually.

## What does managed cloud services include?

Managed cloud services can encompass a variety of IT tasks, including engineering on demand, operations management, 24/7 help desk support, hosting and implementation. By offloading these types of tasks to a trusted partner, you can free up your internal IT teams to focus on more complex initiatives and efforts that drive new business outcomes.

## Benefits of managed cloud services

Partnering with a managed cloud services provider can help organizations achieve numerous goals, both within IT and in the business at large. These outcomes include reduced complexity, improved cost control and accelerated growth and innovation.

### 1.2 IOT Models:

Internet of Things (IoT) bridges the gap between the physical world and the Internet world, making smarter connections possible between machines and everything. No matter for personal use or for diverse vertical industry applications, IoT addresses its significant values in greater convenience, higher efficiency, improved quality of work/production and better utilization of big data. Digital transformation has become the key to business growth, society growth and economic growth, and IoT will be at the center of digital transformation.

The emergence of more high-performance and cost-effective wireless communication technologies, as well as complicated technological specifications, give businesses more IoT opportunities and options to transform their products into future-ready devices. IoT modules, small computing components embedded in devices to enable wireless communication between the device and the network, are the most critical component in an IoT solution. But among diverse module solutions and various manufacturers available on the market, how to choose the optimal module product will be the core of a successful IoT deployment.

## CHAPTER II LITERATURE

### SURVEY

1) **Title:** Vending Machine Technologies: A Review Article, \*

**Author name:** Nilani Ratnasria and Tharaga Sharmilanb

**Year:** 2020

**Content:** Vending Machines are automated machines that dispense selling products such as snacks, beverages, lottery tickets, and etc. It is vital to save time and reduce human energy. These vending machines are developed in the way of Non IoT based and IoT based methods. These Non IoT based machines are not smart and are not operated in real-time data, which are functioned when giving cash or card and inputs (vending things) of the machine. It is controlled by a microcontroller and distributed the given inputs. IoT- based machines are computerized, which have cashless payment facilities, order facility before going to the vending machine to order things, and can be identified the location of machines by the customer. These IoT-based machines are assisted to suppliers to identify the availability of the stocks. Simulation software and prototype are used to validate the machines. In this review, it is found that most of the vending machines developed are capable of operating without IoT technology, and nowadays, vending machine systems are required to implement using IoT with machine learning, and artificial technologies to satisfy the customer preferences.

2) **Title:** Integrated Point of Sales and Snack Vending Machine based on Internet of Things for Self Service Scale Micro Enterprises

**Author name:** R Dijaya

**Year:** 2019

Sales management is a key function which helps small and medium size enterprises (SMEs) in monitoring and tracking stock and co-ordinate transaction processing. Point of Sales developed to support sales management for scale micro enterprises. While self-service applications from banking, to travel, to foodservice, to ticketing, to hotels are booming, many have forgotten that the original self-service applications were in the vending industry. Selfservice has evolved into an operational mode that no longer should be viewed as merely a device-based channel for selling snacks and beverages. The aim of this research integrates point of sales and vending machine based on internet of Things (IoT) for self-service scale micro enterprise. Applications that record while sending data from multiple sensors. Data that records sensors sent through the IoT module to the server, then the server will save the data in a server database. Periodically new data will be visualized in the system through server application is a service. Real-time sales recapitulation system through a vending

machine IoT for 24 hours. After being developed and implemented 24 hours machine will embedded by Internet of Thing (IoT) module. Through this all operational forms of the system are good in the form of stock, recapitulation of sales made by the vending machine and the latest conditions vending machines will be sent to the server so that users can monitor them at any time system. In addition to the profit-taking report, items can be recorded by the system. Owner outlets can remotely controlling vending machines via mobile phones because they are connected to server API (Application Package Interface) to the server application.

## CHAPTER III SYSTEM IMPLEMENTATION

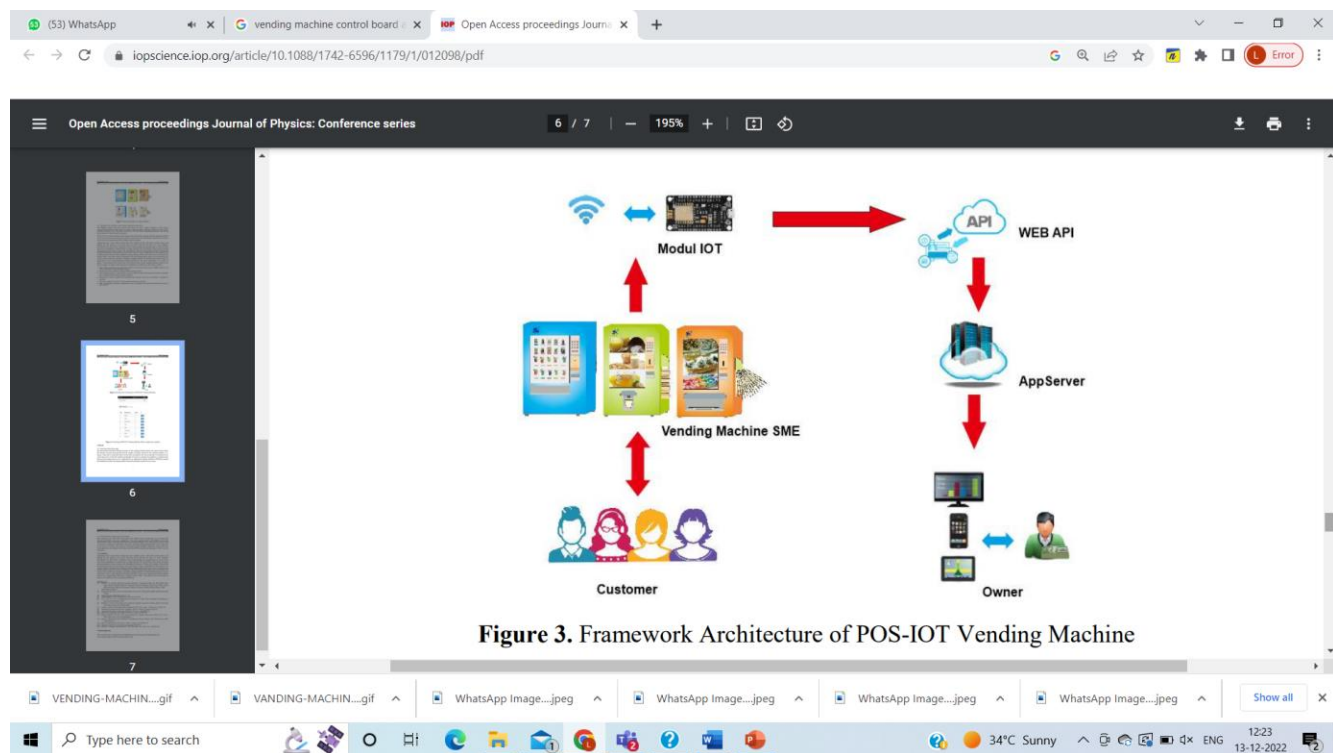
### 3.1 EXISTING SYSTEM

#### History[[edit](#)]

The earliest known reference to a vending machine is in the work of [Hero of Alexandria](#), an engineer, and mathematician in first-century [Roman Egypt](#). His machine accepted a coin and then dispensed [holy water](#).<sup>[2]</sup> When the coin was deposited, it fell upon a pan attached to a lever. The lever opened a valve which let some water flow out. The pan continued to tilt with the weight of the coin until it fell off, at which point a counterweight snapped the lever up and turned off the valve.

Coin-operated machines that dispensed tobacco were being operated as early as 1615 in the [taverns](#) of England. The machines were portable and made of [brass](#).<sup>[3]</sup> An English bookseller, Richard Carlile, devised a newspaper dispensing machine for the dissemination of banned works in 1822. Simon Denham was awarded British Patent no. 706 for his stamp dispensing machine in 1867, the first fully automatic vending machine.<sup>[4]</sup>

### 3.2 PROPOSED SYSTEM



Smart vending machines can be easily distinguished from traditional ones, and are more efficient. Some of the most impressive features of vending machines are as follows:

- **Touchpads and touchless screens** to browse the menu, see the product details, add multiple products to the cart, place an order, etc.
- **Fast and secure transactions:** Smart Vending Machines have digital payment options like UPI, e-Wallet, Debit/Credit Cards, Net Banking, Sodexo, etc.

- **Contactless Ordering:** It was introduced after Covid to provide safer buying options to consumers. It uses wireless communication via a mobile app that Daalchini offers. The transactions are all contactless which has become an important factor post-Covid.
- **Automated Refunds:** In case of failed or canceled orders, the refunds are processed automatically.
- **Remote monitoring:** Retailers can operate the business from anywhere and save money while increasing customer satisfaction. Therefore it increases efficiency.
- **AI-Driven approach:** To sense and deliver personalized experiences, smart vending machines use artificial intelligence and analytics to analyze device data. They can provide the seller with vital customer information.

A smart vending machine has a plethora of benefits and advantages!

- Sensing technology is used to monitor environmental conditions in order to ensure that perishable commodities preserve their quality.
- Data analytics are used to track and manage inventory, and customer data is included in planning to ensure that vending machines are properly stocked.
- They ensure that vending machines do not fall out of service, and maximize their efficiency through remote monitoring and predictive maintenance.
- Any repairs and maintenance can also be done without any hassle.
- To fulfill client demand and expand business growth, get new vending machines and mobile retail sites up and running quickly.
- Improve decision-making and profitability by utilizing real-time data.
- Uses security and authentication to process credit card and debit transactions quickly and securely from any location.
- Built-in security features reduce the danger of tampering and product loss.
- With proper and timely maintenance, these machines have a **life of 8-10 yrs**, and even then if any issues arise they can be dealt with remotely. For this very reason used or refurbished smart vending machines are also a great investment.
- Smart vending machines are now available in the plug & play format which dismissed any assistance during the installation process. The electricity costs are also not very high.
- Smart vending machines can be designed according to your requirements.

### 3.3 DATA COLLECTION

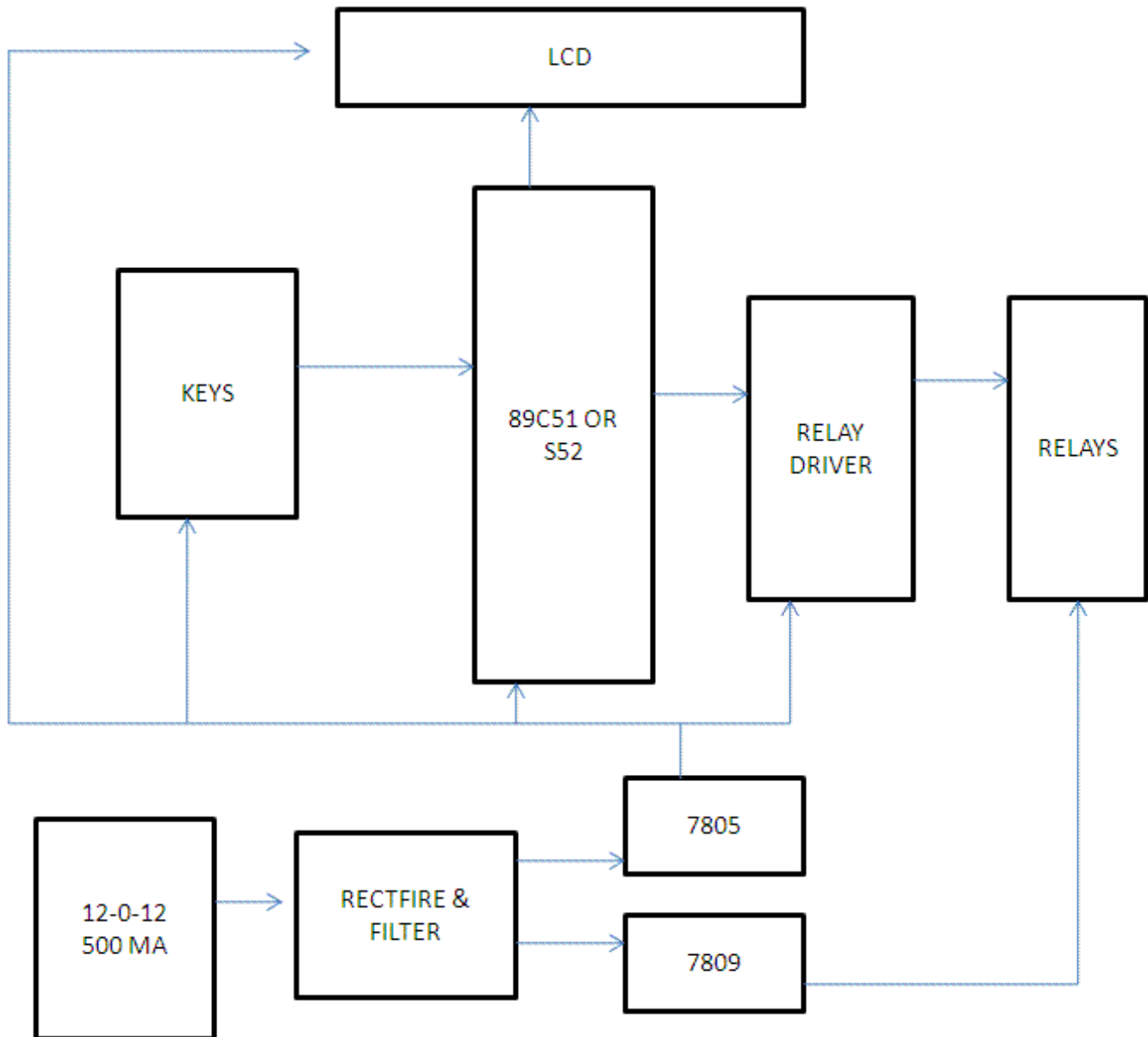
We collected the diabetes disease dataset from kaggle website. This data consists of over 6000 patients data's and has features such as no of praganancies, Glucose level, blood pressure level, skin thickness, Body Mass Index, diabetes pedigree function, age and outcome

### 3.4 DATA PRE-PROCESSING

At first the dataset is fetched by using pandas library and then we save the data inside a pandas data frame, At first this dataset consists of lots of null values, then

we drop all the null values, because our Machine learning model cannot able to process null values

### 3.4.1 PROPOSED SYSTEM ARCHITECTURE





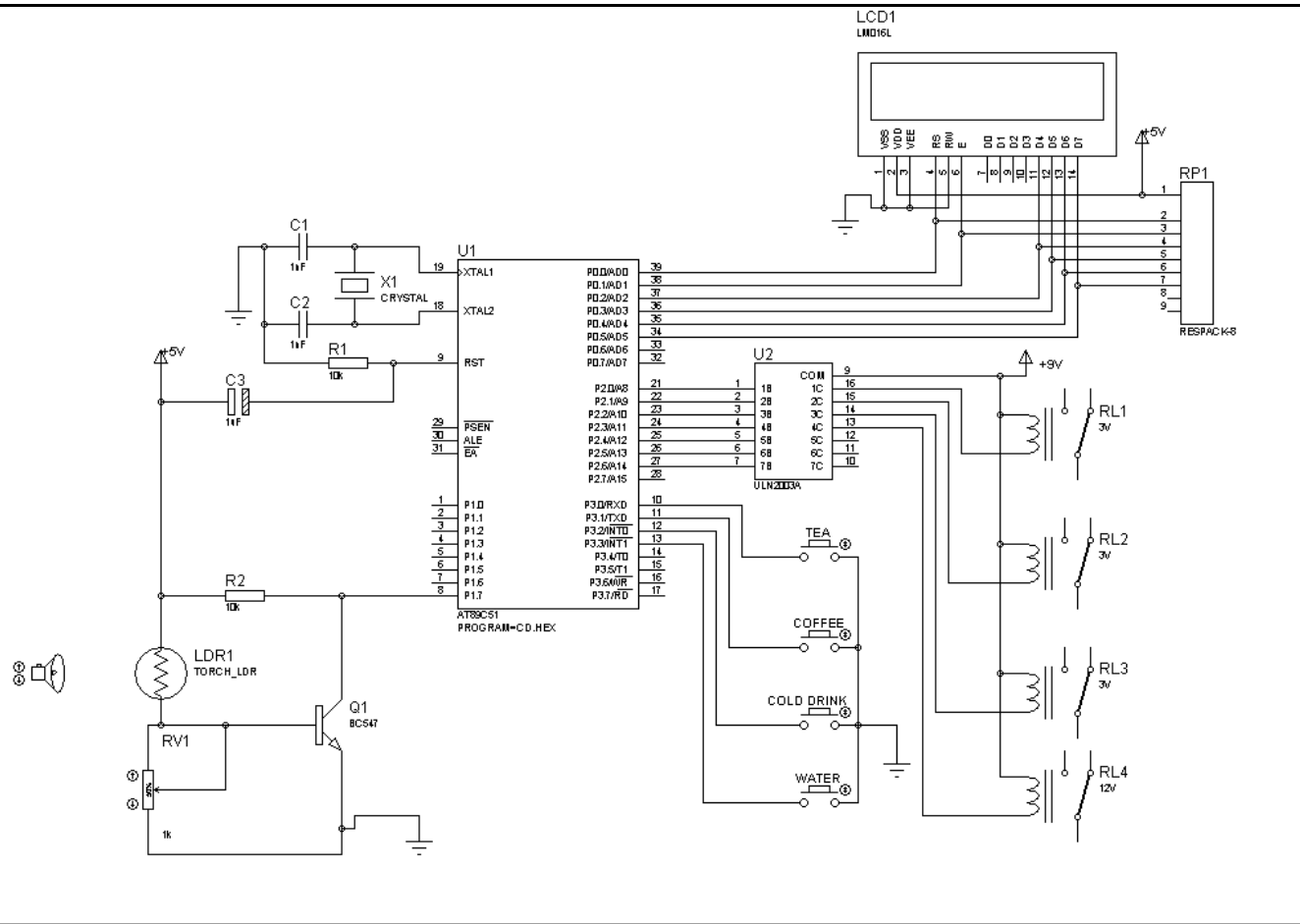


Figure 3.1 Proposed System Architecture

### 3.4.2 MODULE DESCRIPTION

- At first the Diabetes data are driven from the input dataset by using pandas library. Then we pre-process the data by dropping null value, then we make feature selection by selecting input features for feeding it in to the svm, Naïve bayes, decision tree and neural network, the extracted features are inserted in to the machine learning models and the machine gets trained. After training we predict diabetes by feeding test data into the best model. And we make comparisons between performance matrices.

### 3.6 MACHINE LEARNING

Machine learning is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy. IBM has a rich history with machine learning. One of its own, Arthur Samuel, is credited for coining the term, “machine learning” with his research (link resides outside IBM) around the game of checkers. Robert Nealey, the self-proclaimed checkers master, played the game on an IBM 7094 computer in 1962, and he lost to the computer. Compared to what can be done today, this feat almost seems trivial, but it’s considered a major milestone within the field of artificial intelligence. Over the next couple of decades, the technological developments around storage and processing power will enable some innovative products that we know and love today, such as Netflix’s recommendation engine or self-driving cars. Machine learning is an important component of the growing field of data science. Through the use of statistical methods, algorithms are trained to make classifications or predictions, uncovering key insights within data mining projects. These insights subsequently drive decision making within applications and businesses, ideally impacting key growth metrics. As big data continues to expand and grow, the market demand for data scientists will increase, requiring them to assist in the identification of the most relevant business questions and subsequently the data to answer them.

## CHAPTER IV

### SYSTEM DESIGN & IMPLEMENTATIONS

#### 4.1 SOFTWARE DESCRIPTION

The Python language had a humble beginning in the late 1980s when a Dutchman Guido Von Rossum started working on a fun project, which would be a successor to ABC language with better exception handling and capability to interface with OS Amoeba at Centrum Wiskunde and Informatica. It first appeared in 1991. Python 2.0 was released in the year 2000 and Python 3.0 was released in the year 2008. The language was named Python after the famous British television comedy show Monty Python's Flying Circus, which was one of Guido's favorite television programmes. Here we will see why Python has suddenly influenced our lives and the various applications that use Python and its implementations.

In this chapter, you will be learning the basic installation steps that are required to perform on different platforms (that is Windows, Linux, and Mac), about environment variables, setting up of environment variables, file formats, Python interactive shell, basic syntaxes and finally printing out formatted output.

##### 4.1.1 PYTHON

Now you might be suddenly bogged with the question, why Python? According to Institute of Electrical and Electronics Engineers (IEEE) 2016 ranking Python ranked third after C and Java. As per Indeed.com's data of 2016, the Python job market search ranked fifth. Clearly, all the data points to the ever rising demand in the job market for Python. It's a cool language if you want to learn just for fun or if you want to build your career around Python, you will adore the language. At school level, many schools have started including Python programming for kids. With new technologies taking the market by surprise Python has been playing a dominant role. Whether it is cloud

platform, mobile app development, Big Data, IoT with Raspberry Pi, or the new Block chain technology, Python is being seen as a niche language platform to develop and deliver a scalable and robust applications.

Some key features of the language are:

- Python programs can run on any platform, you can carry code created in Windows machine and run it on Mac or Linux
- Python has inbuilt large library with prebuilt and portable functionality, also known as the standard library
- Python is an expressive language
- Python is free and open source
- Python code is about one third of the size of equivalent C++ and Java code
- Python can be both dynamically and strongly typed--dynamically typed means it is a type of variable that is interpreted at runtime, which means, in Python, there is no need to define the type (int or float) of the variable

## ● PYTHON APPLICATIONS

One of the most famous platforms where Python is extensively used is YouTube. The other places where you will find Python being extensively used are the special effects in Hollywood movies, drug evolution and discovery, traffic control systems, ERP systems, cloud hosting, e-commerce platform, CRM systems, and whatever field you can think of.

## ● VERSIONS

At the time of writing this book, two main versions of the Python programming language were available in the market, which are Python 2.x and Python 3.x. The stable release as of writing the book were Python 2.7.13 and Python 3.6.0.

## ● IMPLEMENTATIONS OF PYTHON

Major implementations include CPython, Jython, IronPython, MicroPython, and PyPy.

#### 4.1.2 INSTALLATION

Here we will look forward to the installation of Python on three different OS platforms, namely, Windows, Linux, and Mac OS. Let's begin with the Windows platform.

- INSTALLATION ON WINDOWS PLATFORM

Python 2.x can be downloaded from <https://www.python.org/downloads>. The installer is simple and easy to install. Perform the following steps to install the setup:

Step 1. Once you click on setup installer, you will get a small window on your desktop screen as shown here; click on **Next**:



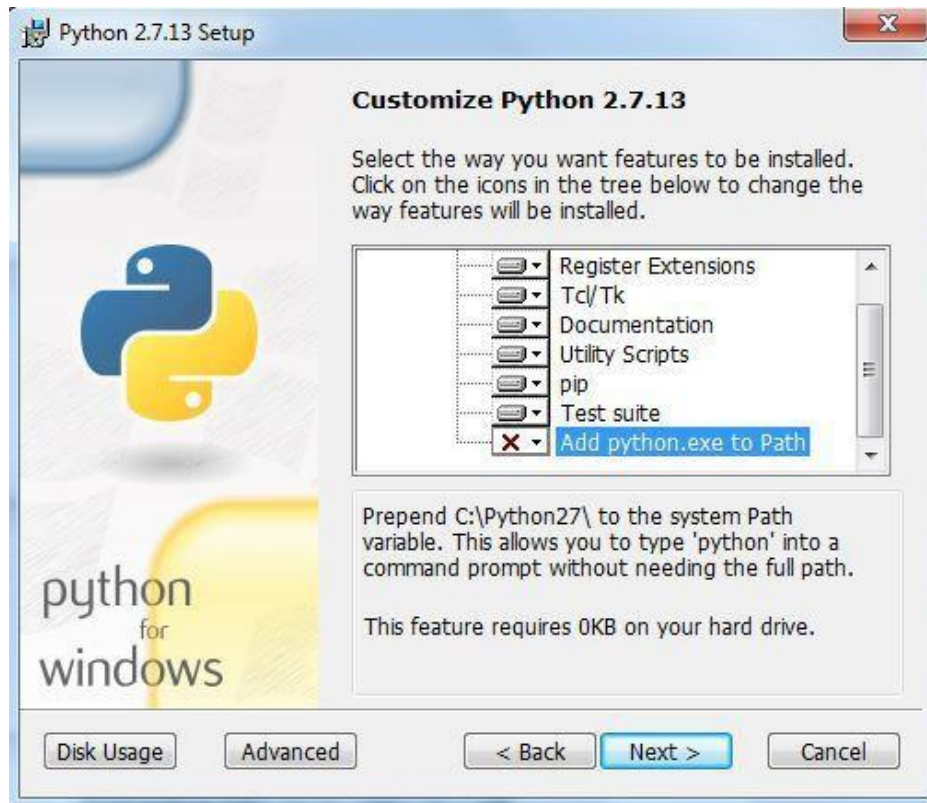
### Figure 4.1 Graphical interface of Python Installation

Step 2. Provide a suitable installation folder to install Python. If you don't provide the installation folder, then the installer will automatically create an installation folder for you, as shown in the following screenshot. Click on **Next**:



Figure 4.2 Graphical interface of Python Destination Directory

Step 3. After completion of step 2, you will get a window to customize Python as shown in the preceding screenshot. Notice that the **Add python.exe to Path** option has been marked **x**. Select this option to add it to system path variable (which will be explained later in the chapter), and click on **Next**:



**Figure 4.3 Graphical interface of Python Customization**



### Figure 4.4 Graphical interface of Python instalization Completed

Step 4. Finally, click on **Finish** to complete the installation:

#### 4.1.3 PYTHON FILE FORMATS

Every language understands a file format, for example, like the C language file extension is .c likewise java language has a file extension .java. The Python fileextension is .py while bytecode file extension is .pyc.

- **PYTHON INTERACTIVE SHELL**

Python interactive shell is also known as **Integrated Development Environment (IDLE)**. With the Python installer, two interactive shells are provided: one is IDLE (Python GUI) and the other is Python (command line). Both can be used for running simple programs. For complex programs and executing large files, the windows command prompt is used, where after the system variables are set automatically, large files are recognized and executed by the system.

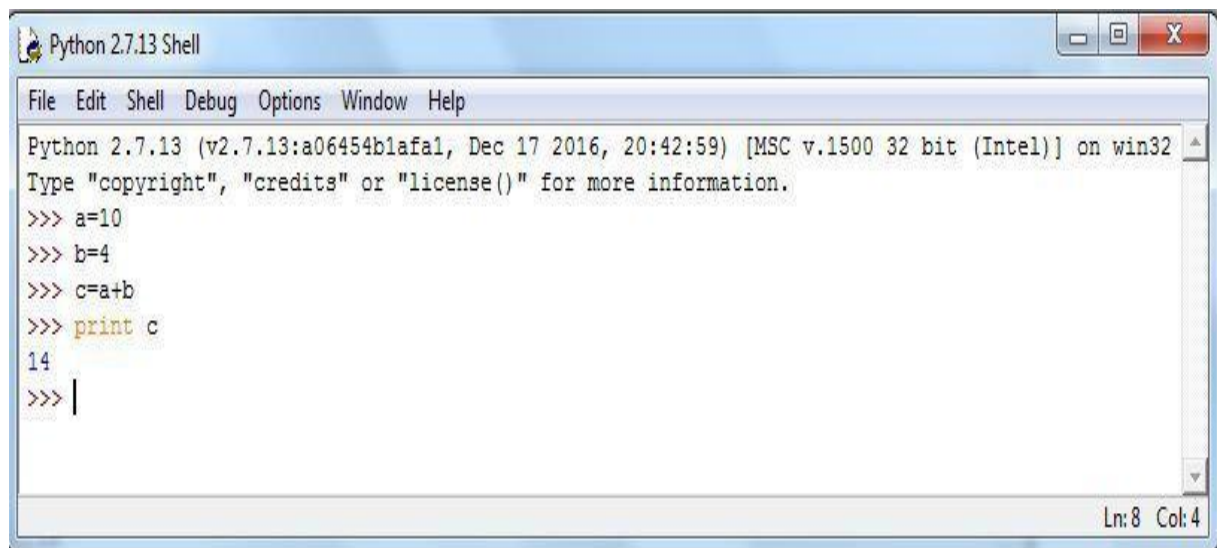
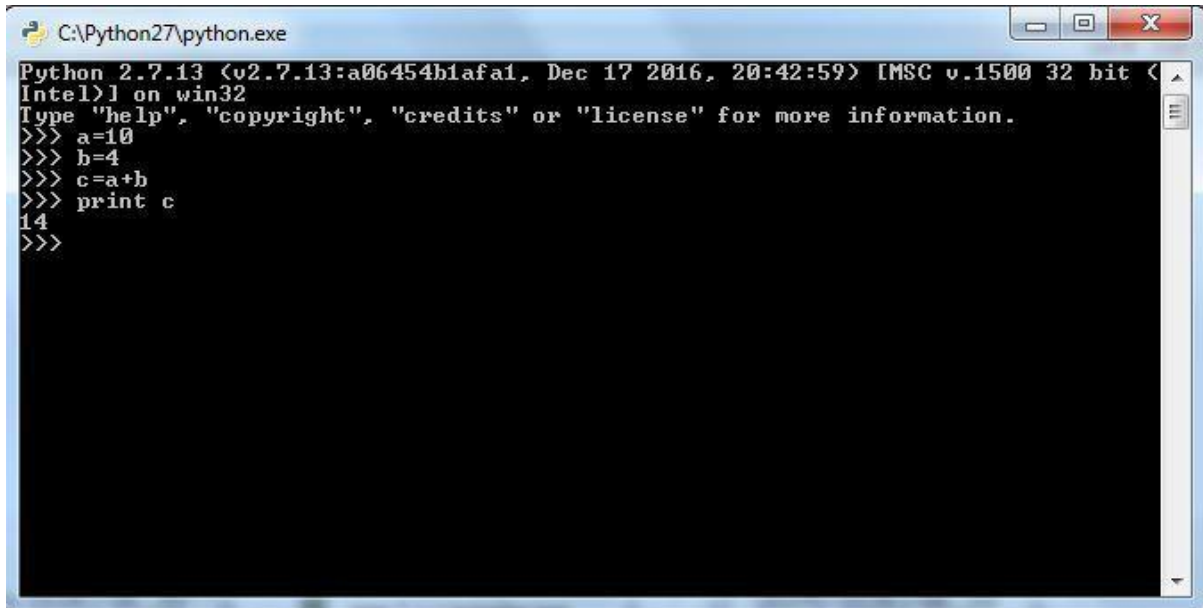


Figure 4.5 Graphical interface of Python interactive Shell

The preceding screenshot is what we call Python IDLE, which comes bundled with the Python installation. The next screenshot is of the command line that also



comes bundled with the Python installation, or we can simply launch the Python command through the windows command line and get Python command line. For most of our programming instructions, we will be using the Python command line:



**Figure 4.6 Graphical interface of Python Execution**

#### 4.1.4 SYNTAX AND SEMANTICS

Python is meant to be an easily readable language. Its formatting is visually uncluttered, and it often uses English keywords where other languages use punctuation. Unlike many other languages, it does not use curly brackets to delimit blocks, and semicolons after statements are optional. It has fewer syntactic exceptions and special cases than C or Pascal.

- **INDENTATION**

Python uses whitespace indentation, rather than curly brackets or keywords, to delimit blocks. An increase in indentation comes after certain statements; a decrease in

indentation signifies the end of the current block. Thus, the program's visual structure accurately represents the program's semantic structure. This feature is sometimes termed the off-side rule, which some other languages share, but in most languages indentation doesn't have any semantic meaning.

## ● STATEMENTS AND CONTROL FLOW

Python's statements include (among others):

- The **assignment** statement (token '=', the equals sign). This operates differently than in traditional imperative programming languages, and this fundamental mechanism (including the nature of Python's version of variables) illuminates many other features of the language. Assignment in C, e.g.,  $x = 2$ , translates to "typed variable name  $x$  receives a copy of numeric value 2". The (right-hand) value is copied into an allocated storage location for which the (left-hand) variable name is the symbolic address. The memory allocated to the variable is large enough (potentially quite large) for the declared type. In the simplest case of Python assignment, using the same example,  $x = 2$ , translates to "(generic) name  $x$  receives a reference to a separate, dynamically allocated object of numeric (int) type of value 2." This is termed binding the name to the object. Since the name's storage location doesn't contain the indicated value, it is improper to call it a variable. Names may be subsequently rebound at any time to objects of greatly varying types, including strings, procedures, complex objects with data and methods, etc. Successive assignments of a common value to multiple names, e.g.,  $x = 2$ ;  $y = 2$ ;  $z = 2$  result in allocating storage to (at most) three names and one numeric object, to which all three names are bound. Since a name is a generic reference holder it is unreasonable to associate a fixed data type with it. However at a given time a

name will be bound to some object, which will have a type; thus there is dynamic typing.

- The **if** statement, which conditionally executes a block of code, along with else and elif (a contraction of else-if).
- The **for** statement, which iterates over an iterable object, capturing each element to a local variable for use by the attached block.
- The **while** statement, which executes a block of code as long as its condition is true.
- The **try** statement, which allows exceptions raised in its attached code block to be caught and handled by except clauses; it also ensures that clean-up code in a finally block will always be run regardless of how the block exits.
- The **raise** statement, used to raise a specified exception or re-raise a caught exception.
- The **class** statement, which executes a block of code and attaches its local namespace to a class, for use in object-oriented programming.
- The **def** statement, which defines a function or method.
- The **with** statement, from Python 2.5 released in September 2006, which encloses a code block within a context manager (for example, acquiring a lock before the block of code is run and releasing the lock afterwards, or opening a file and then closing it), allowing Resource Acquisition Is Initialization (RAII)-like behavior and replaces a common try/finally idiom.
- The **break** statement, exits from the loop.
- The **continue** statement, skips this iteration and continues with the next item.
- The **pass** statement, which serves as a NOP. It is syntactically needed to create an empty code block.
- The **assert** statement, used during debugging to check for conditions that ought to apply.

- The **yield** statement, which returns a value from a generator function. From Python 2.5, yield is also an operator. This form is used to implement coroutines.
- The **import** statement, which is used to import modules whose functions or variables can be used in the current program. There are three ways of using import: import <module name> [as <alias>] or from <module name> import \* or from <module name> import <definition 1> [as <alias 1>], <definition 2> [as <alias 2>], ....
- The **print** statement was changed to the print() function in Python 3.

#### 4.1.5 PYTHON PROGRAMMING EXAMPLES

Hello world program:

```
print('Hello, world!')
```

Program to calculate the factorial of a positive integer:

```
n = int(input('Type a number, then its factorial will be printed: '))if n <
```

```
0:
```

```
raise ValueError('You must enter a positive number')fact =
```

```
1
```

```
i = 2 while i <= n:
```

```
fact = fact * i i += 1
```

```
print(fact)
```

## What is Amazon EC2?

Amazon EC2 is a cloud computing platform that can be auto-scaled to meet demand.

Different hardware and software configurations can be selected. Different geographical locations can be selected be closer to users, as well as providing redundancy in case of failures.

Persistent storage can be provided by Amazon EBS (Elastic Block Storage). [Amazon S3 \(Simple Storage Service\)](#) data can also be accessed with Amazon EC2 instances, and is free if they are in the same region.

# What is AWS RDS?

- Amazon RDS is a Relational Database Cloud Service
- Amazon RDS minimizes relational database management by automation
- Amazon RDS creates multiple instances for high availability and failovers
- Amazon RDS supports PostgreSQL, MySQL, Maria DB, Oracle, SQL Server, and Amazon Aurora

## Relational Databases

- Amazon RDS stores data as tables, records, and fields
- Values from one table can have a relationship to values in other tables. Relationships are a key part of relational databases.
- Relational databases are often used for storing transactional and analytical data
- Relational databases provide stability and reliability for transactional databases

## What is AWS S3?

Amazon S3 (Simple Storage Service) provides object storage, which is built for storing and recovering any amount of information or data from anywhere over the internet. It provides this storage through a web services interface. While designed for developers for easier web-scale computing, it provides 99.999999999 percent durability and 99.99 percent availability of objects. It can also store computer files up to 5 terabytes in size.

## AWS S3 Benefits

Some of the benefits of AWS S3 are:

- **Durability:** S3 provides 99.999999999 percent durability.
- **Low cost:** S3 lets you store data in a range of “storage classes.” These classes are based on the frequency and immediacy you require in accessing files.
- **Scalability:** S3 charges you only for what resources you actually use, and there are no hidden fees or overage charges. You can scale your storage resources to easily meet your organization’s ever-changing demands.
- **Availability:** S3 offers 99.99 percent availability of objects

- Security: S3 offers an impressive range of access management tools and encryption features that provide top-notch security.
- Flexibility: S3 is ideal for a wide range of uses like data storage, data backup, software delivery, data archiving, disaster recovery, website hosting, mobile applications, IoT devices, and much more.
- Simple data transfer: You don't have to be an IT genius to execute data transfers on S3. The service revolves around simplicity and ease of use.

These are compelling reasons to sign up for S3. Now, let's move on and have a look at some of the major components of the AWS S3 storage service.

## AWS Buckets and Objects

An object consists of data, key (assigned name), and metadata. A bucket is used to [store objects](#). When data is added to a bucket, Amazon S3 creates a unique version ID and allocates it to the object.

### Jenkins

Jenkins is an open source [continuous integration](#)/continuous delivery and deployment (CI/CD) automation software DevOps tool written in the [Java](#) programming language. It is used to implement CI/CD workflows, called pipelines.

Pipelines automate testing and reporting on isolated changes in a larger code base in real time and facilitates the integration of disparate branches of the code into a main branch. They also rapidly detect defects in a code base, build the software, [automate testing](#) of their builds, prepare the code base for deployment (delivery), and ultimately deploy code to containers and virtual machines, as well as bare metal and cloud servers. There are several commercial versions of Jenkins. This definition only describes the upstream [open source](#) project.

## Jenkins and CI/CD

Over time, [continuous delivery](#) and deployment features have been added to Jenkins. Continuous delivery is the process of automating the building and packaging of code for eventual deployment to test, production staging, and production environments. Continuous deployment automates the final step of deploying the code to its final destination.

In both cases, automation reduces the number of errors that occur because the correct steps and best practices are encoded into Jenkins. Jenkins describes a desired state and the automation server

ensures that that state is achieved. In addition, the velocity of releases can be increased since deployments are no longer bounded by personnel limitations, such as operator availability. Finally, Jenkins reduces stress on the development and operations team, by removing the need for middle of the night and weekend rollouts.

## Jenkins and microservices

The need for Jenkins becomes especially acute when deploying to a [microservices](#) architecture. Since one of the goals of microservices is to frequently update applications and services, the ability to do so cannot be bounded by release bandwidth. More and smaller services with faster update intervals can only be achieved by the type of automation Jenkins provides.

## Jenkins X

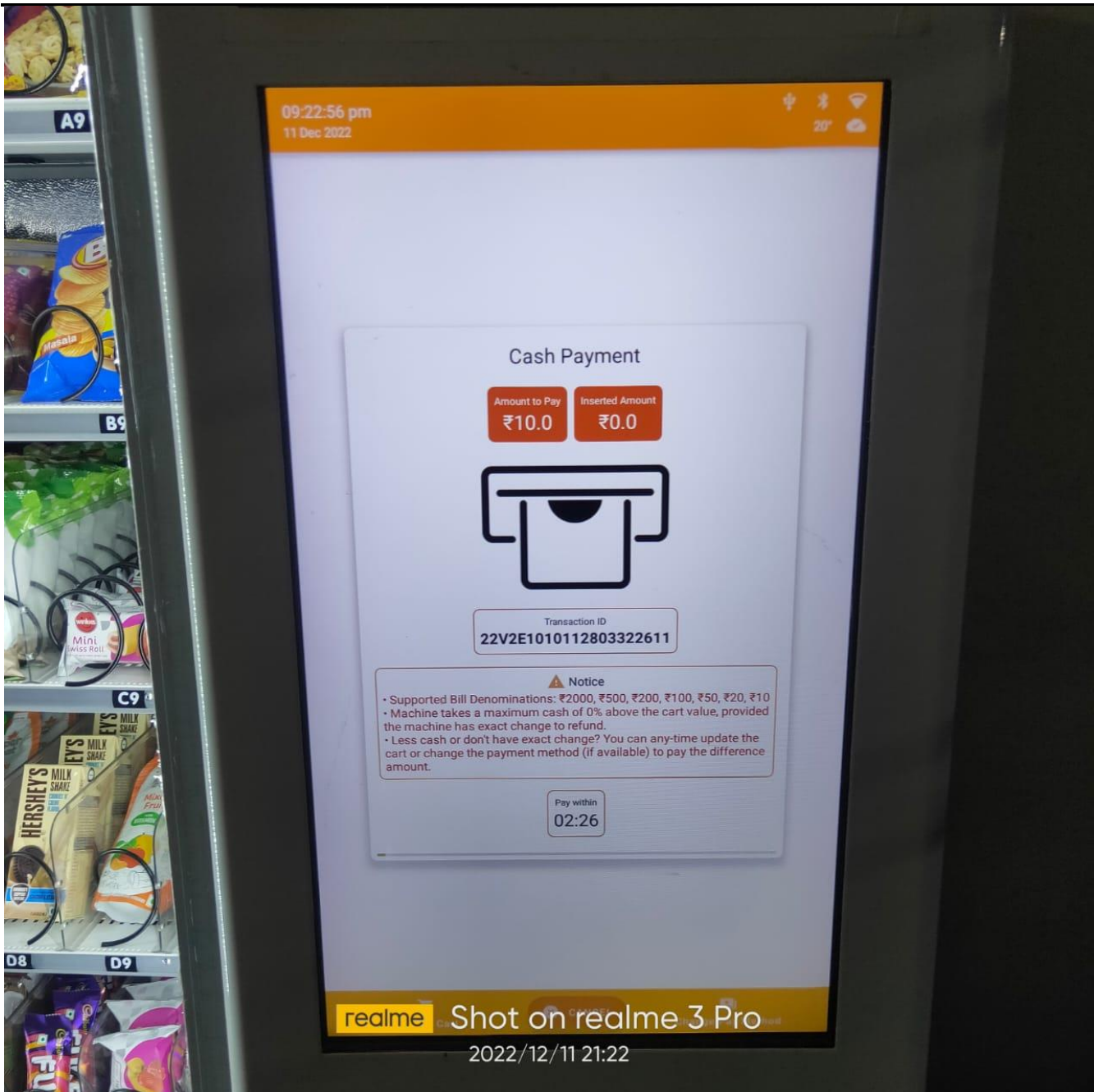
The [Jenkins X](#) project was formerly launched in 2018 with the goal of creating a modern, cloud native Jenkins. It is also a project under the guidance of the CD Foundation. Its architecture, technology and pipeline language are completely different from Jenkins. Jenkins X is designed for [Kubernetes](#) and uses it in its own implementation. Other cloud native technology that Jenkins X uses are Helm and Tekton.

## How Jenkins works

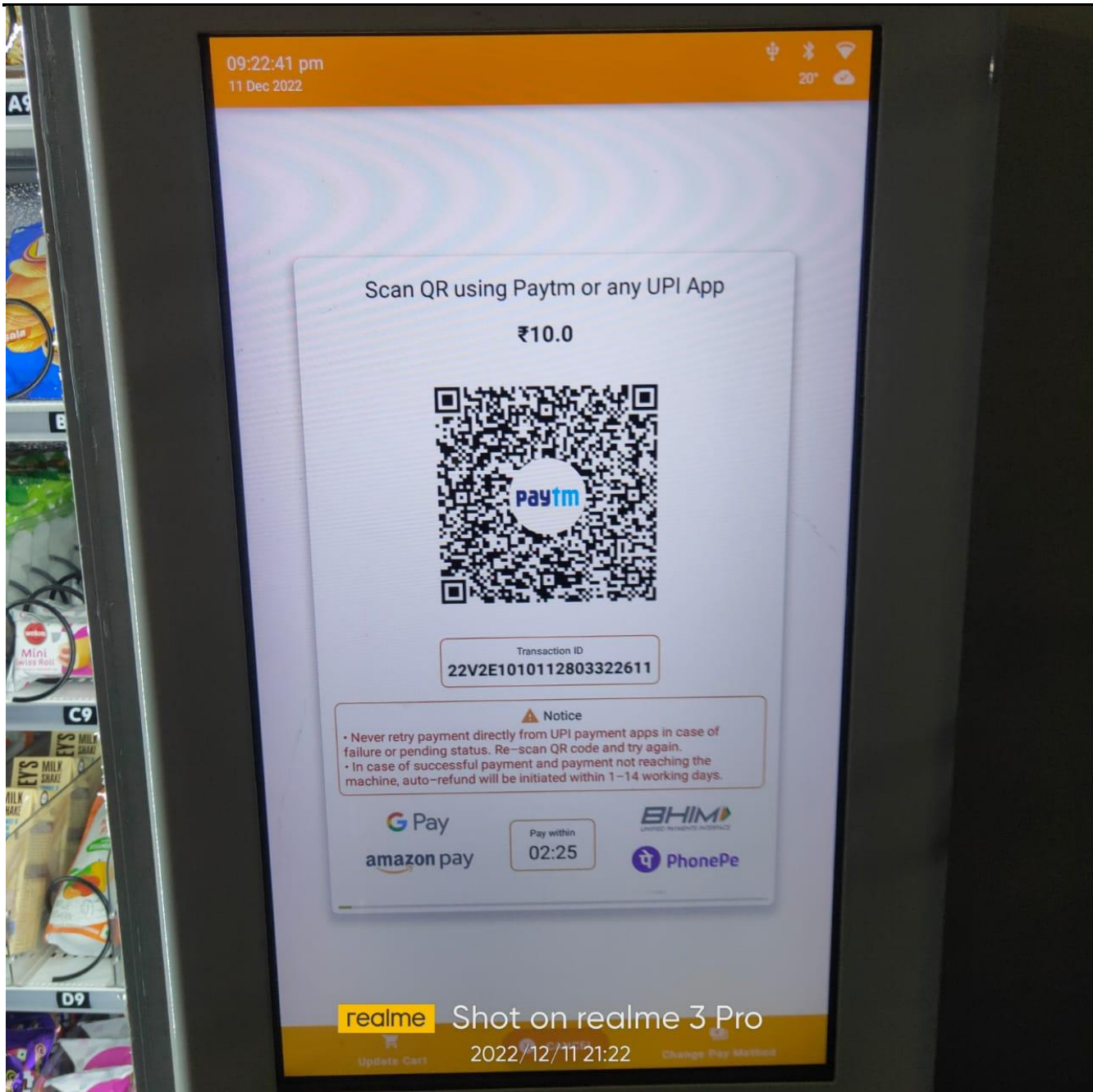
Jenkins runs as a server on a variety of platforms including Windows, MacOS, Unix variants and especially, [Linux](#). It requires a Java 8 VM and above and can be run on the Oracle JRE or [OpenJDK](#). Usually, Jenkins runs as a Java servlet within a Jetty application server. It can be run on other Java application servers such as Apache Tomcat. More recently, Jenkins has been adapted to run in a [Docker container](#). There are read-only Jenkins images available in the Docker Hub online repository.

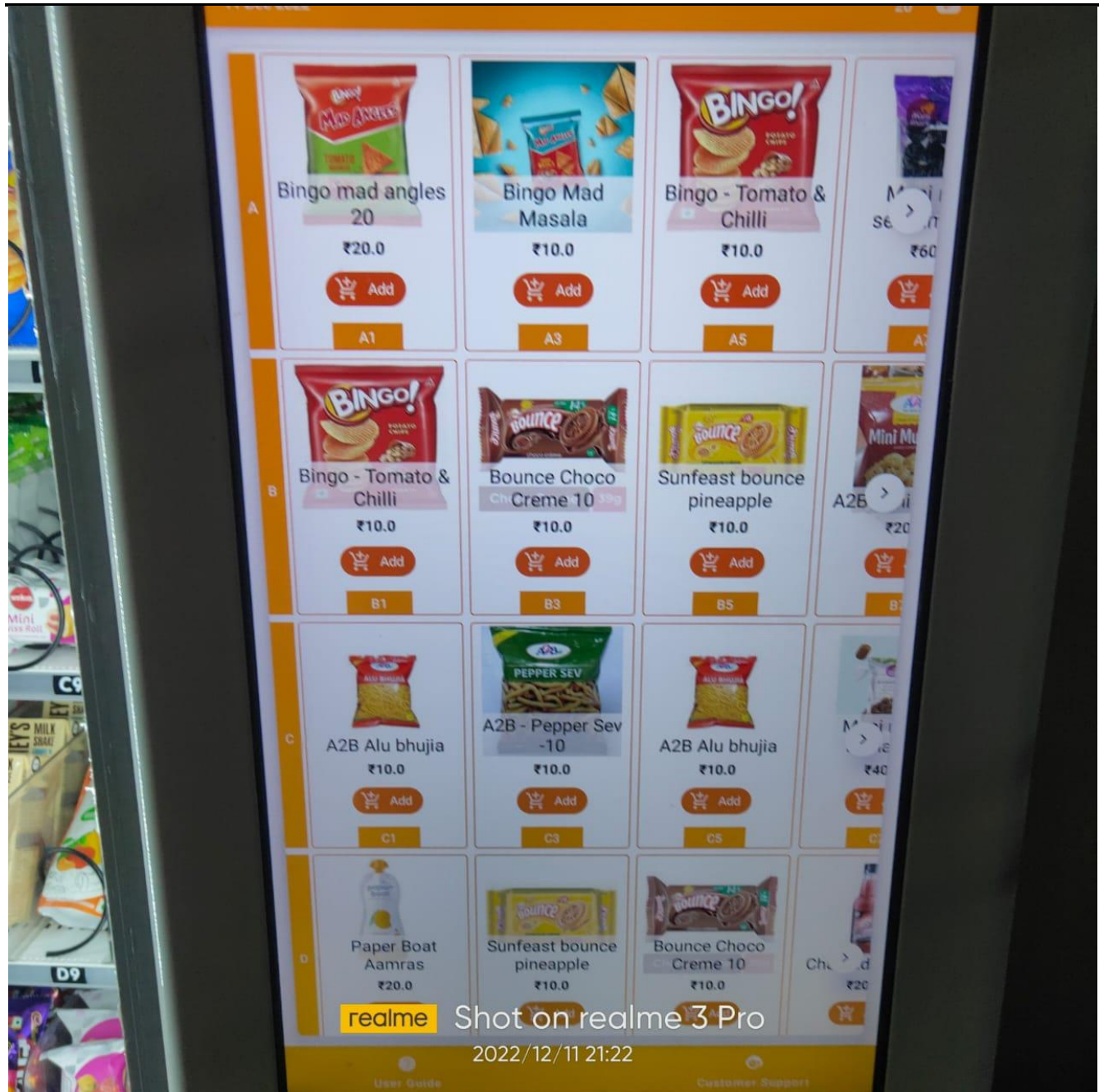
To operate Jenkins, pipelines are created. A pipeline is a series of steps the Jenkins server will take to perform the required tasks of the CI/CD process. These are stored in a plain text *Jenkinsfile*. The *Jenkinsfile* uses a curly bracket syntax that looks similar to [JSON](#). Steps in the pipeline are declared as commands with parameters and encapsulated in curly brackets. The Jenkins server then reads the *Jenkinsfile* and executes its commands, pushing the code down the pipeline from committed source code to production runtime. A *Jenkinsfile* can be created through a GUI or by writing code directly.

<https://fournex.riota.in/home/login>

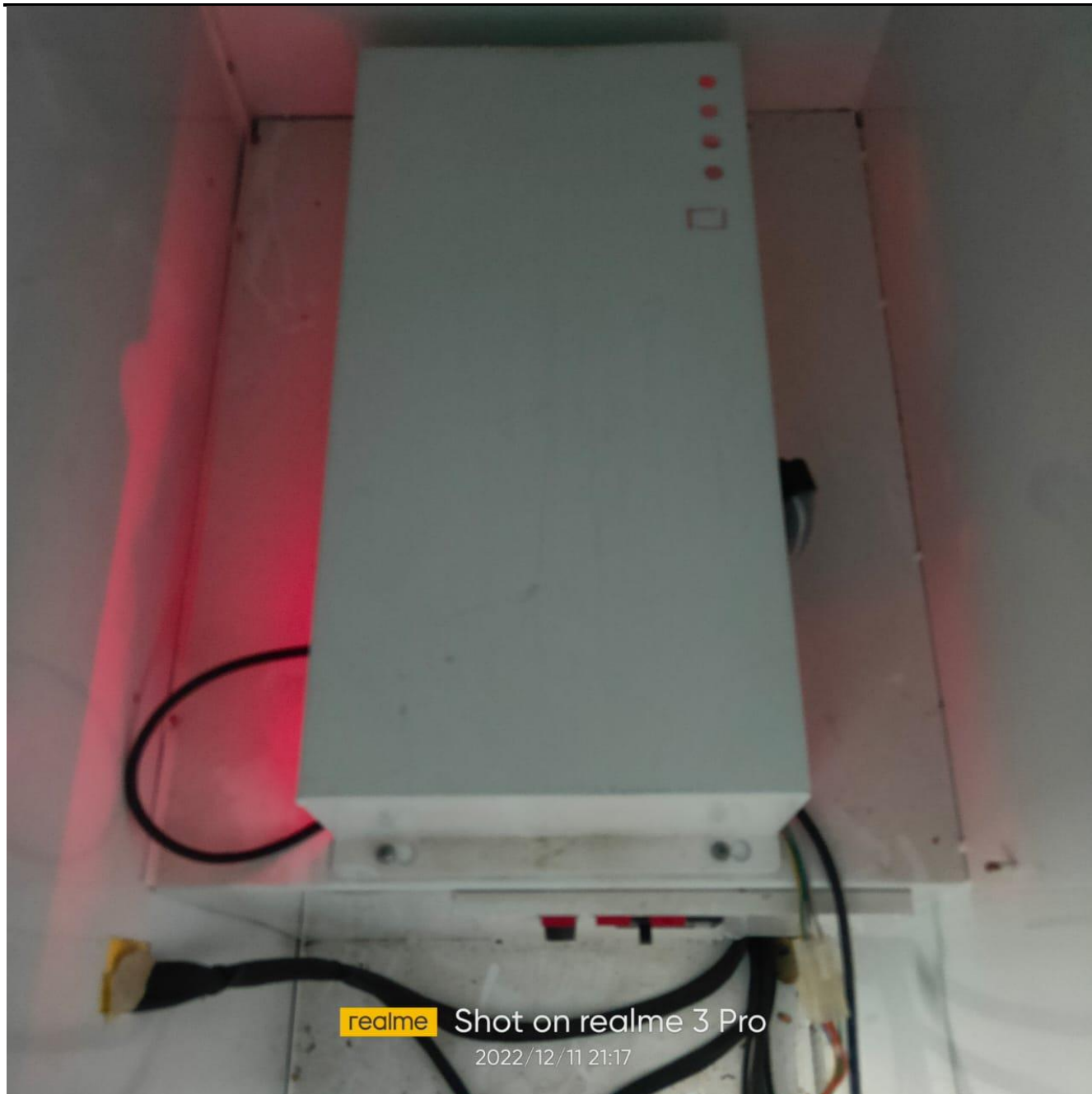


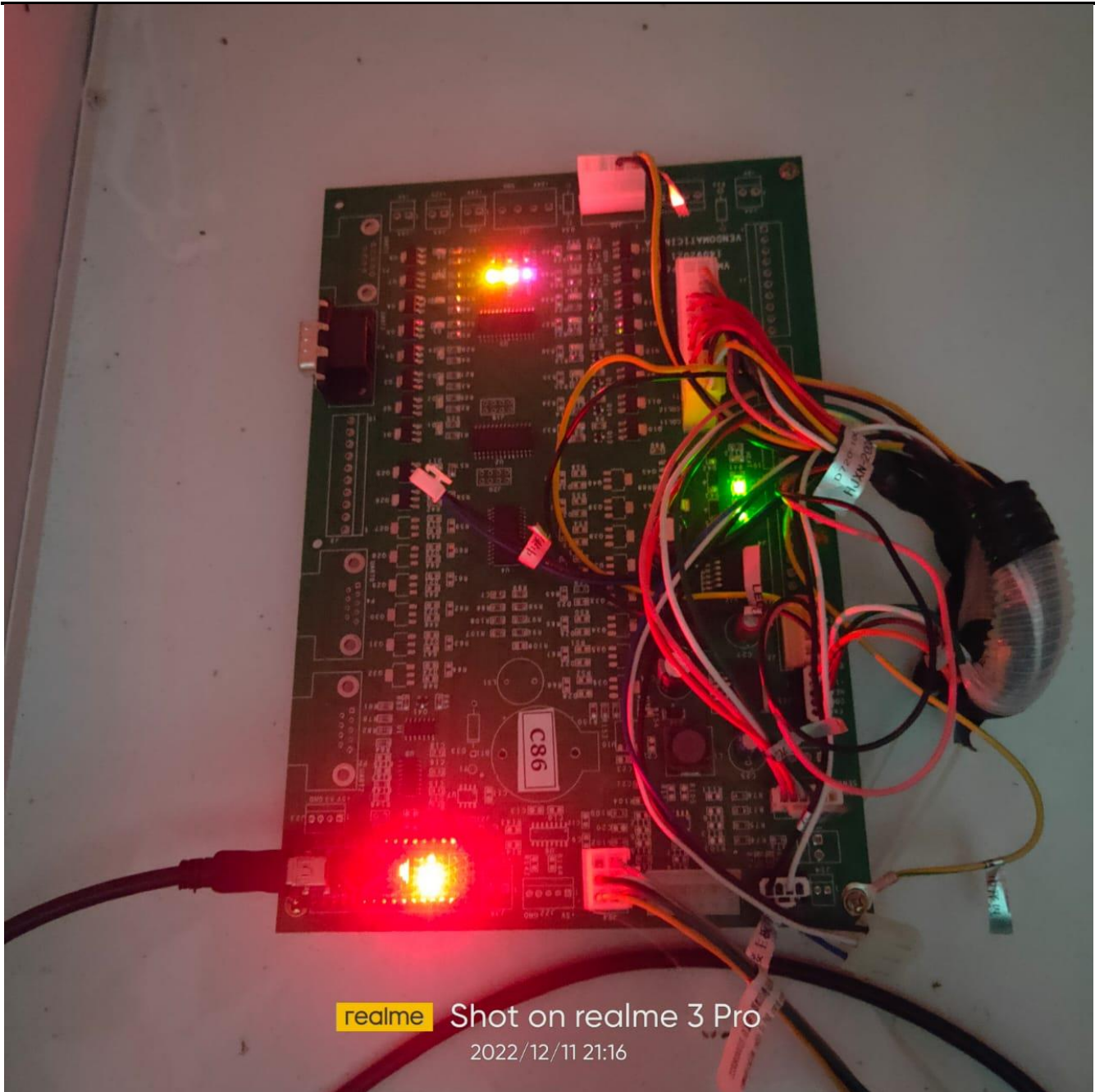




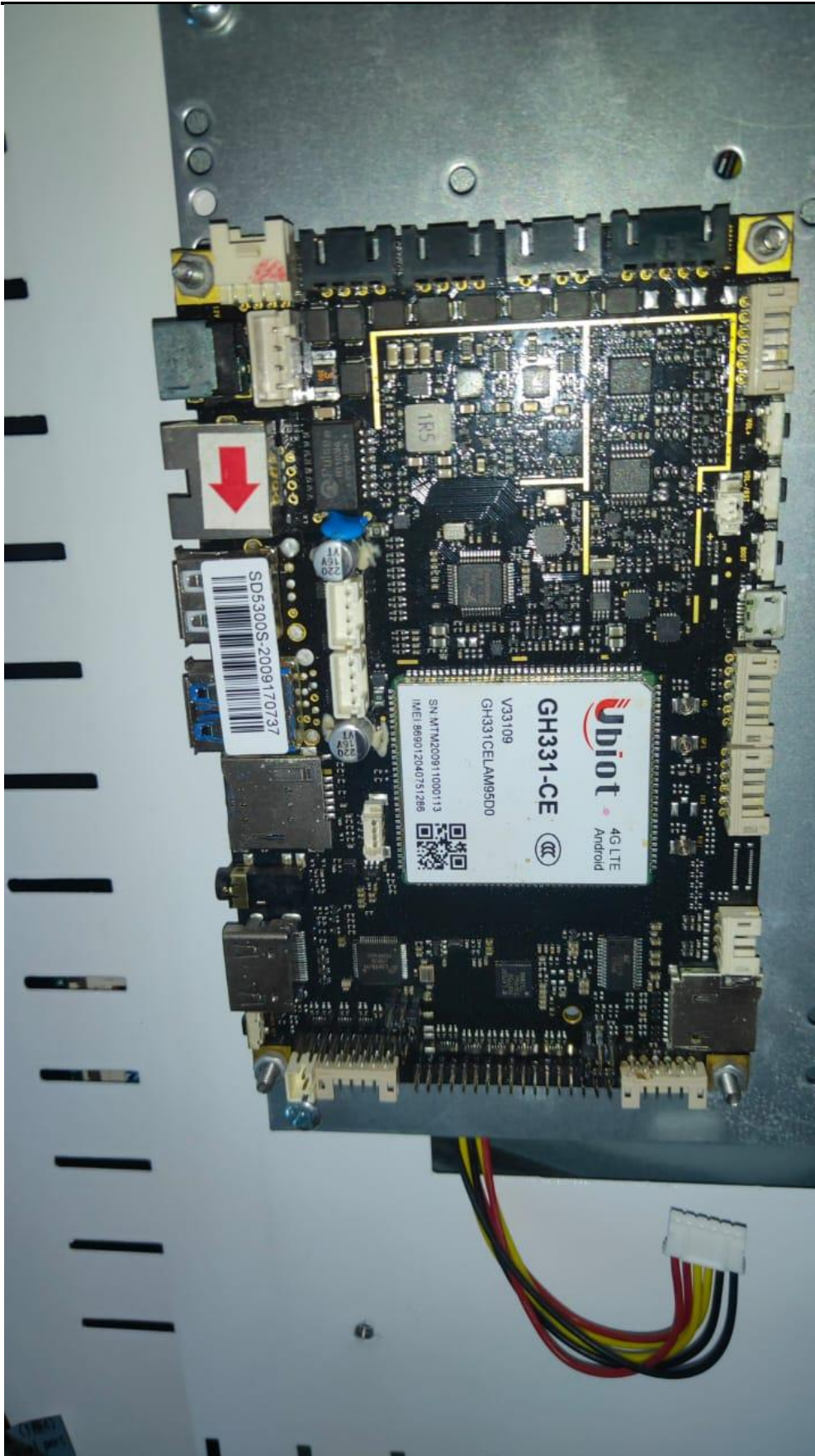








realme Shot on realme 3 Pro  
2022/12/11 21:16



fournex.riota.in/portal/home

Inbox - omguruco... நில் உரிமை (ப... Inspector General o... Google WhatsApp Web iLovePDF | Online P... வட்டாட்சியர் அ... Crop and Print Aad...

**FOURNEX Group** Welcome, Fournex Portal

Dashboard

- Machines >
- Transactions >
- Products >
- Warehouse >
- W. Transactions >
- Advertisements >
- User Management >
- API Integration
- Company >
- Purchase Order >
- Offers & Coupons >
- Help & Support

**Machines**  
54

**Clients**  
40

**Active Machines**  
49

**Inactive Machines**  
5

**Company Users**  
36

**Client Users**  
0

**Transactions**  
49031

**Sales**  
1548325

**Payment**

Cash	0.00	Cashless	1597085.00
Credit Card (Ezetap)	482870.00	Credit Card (Razorpay)	0.00
Refund	7135.00	Refund Processed	0.00
<b>Total</b>		<b>:1597085.00</b>	

**Machine Operation Status**

Online	49	Offline	0
Down (Planned)	0	Down	3
Standby	0	Terminated	2
N/A	0		

**Machine Stock Status**

75 - 100%	50 - 75%	25 - 50%	0 - 25%	N/A
0	0	0	2	52

**Product Sales**

Date Range: 11/12/2022 ~ 12/12/2022
Time: Daily
Type: Product

Copyright © RIOTA Private Limited 2022 Version: 8.2.3

barath copy.docx | barath copy.pdf | 170226040533560...jpg

Type here to search | 28°C | 2:04 PM | 12/12/2022

- Dashboard
- Machines
- Machines List
- Current Stock
- Requests
- Machine Config
- Transactions
- Products
- Warehouse
- W. Transactions
- Advertisements
- User Management
- API Integration
- Company

Machines List

Search Type: Machine Id    Search Text:    Sort By: Id

Id	Machine Id	Machine Model	Machine Type	Client Id	Client Name	Branch Id	Branch Name	Operation Status	Connection Status	Created On
18454	2VE0018454	VENT5SWIDE	22 Inch Sprial Combo	CL4200	Rathinam kalam block	BR01017	JP Trichy	Online	Connected	2022/12/03 15:04:13
18453	2VE0018453	VENT5SWIDE	22 Inch Sprial Combo	CL4199	capegemini salem	BR01017	JP Trichy	Online	Connected	2022/12/02 18:28:21
18452	2VE0018452	VENT22INCS	22 Inch Spring	CL4198	BILEDRIGH T Trichy	BR01017	JP Trichy	Online	Connected	2022/11/21 12:18:09
18451	2VE0018451	VENT22INCS	22 Inch Spring					Online	Disconnected	2022/11/20 18:03:03
18450	2VE0018450	VENT22INCS	22 Inch Spring					Online	Disconnected	2022/11/20 18:01:47
18449	2VE0018449	VENT22INCS	22 Inch Spring					Online	Disconnected	2022/11/20 18:00:40
18448	2VE0018448	VENT22INCS	22 Inch Spring					Online	Disconnected	2022/11/20 17:49:58
18447	2VE0018447	VENT5SWIDE	22 Inch Sprial Combo					Online	Disconnected	2022/11/20 17:47:14
18446	2VE0018446	VENT22INCS	22 Inch Spring					Online	Disconnected	2022/11/07 12:21:24
18445	2VE0018445	VENT22INCS	22 Inch Spring					Online	Disconnected	2022/11/07 12:21:24

Items per page: 10    1 - 10 of 54



fournex.riota.in/portal/machine-info

Inbox - omguruo... நில் உரிமை (ப... Inspector General o... Google WhatsApp Web iLovePDF | Online P... வட்டாரமியர் அ... Crop and Print Aad...

Welcome, Fournex Portal

- Dashboard
- Machines >
  - Machines List
  - Current Stock
  - Requests
  - Machine Config
- Transactions >
- Products >
- Warehouse >
- W. Transactions >
- Advertisements >
- User Management >
- API Integration
- Company >

Row	Machine ID	Status	Product	Units	Price
Row 1	A x 1	Enabled	Bingo mad angles 20	0 Units	- 20 ₹
	A x 2	Disabled	N/A	0 Units	- N/A ₹
	A x 3	Enabled	Bingo mad angles 20	0 Units	- 20 ₹
	A x 4	Disabled	N/A	0 Units	- N/A ₹
Row 2	B x 1	Enabled	BINGO - Chips Masala-	0 Units	- 10 ₹
	B x 2	Disabled	N/A	0 Units	- N/A ₹
	B x 3	Enabled	Lays - Classic Salted	0 Units	- 10 ₹
	B x 4	Disabled	N/A	0 Units	- N/A ₹
Row 3	C x 1	Enabled	A2B - Moong Dal -10	0 Units	- 10 ₹
	C x 2	Disabled	N/A	0 Units	- N/A ₹
	C x 3	Enabled	A2B Om Podi	0 Units	- 10 ₹
	C x 4	Disabled	N/A	0 Units	- N/A ₹
Row 4	D x 1	Enabled			
	D x 2	Enabled			
	D x 3	Enabled			
	D x 4	Enabled			

Copyright © RIOTA Private Limited 2022 Version: 8.2.3

barath copy.docx barath copy.pdf 170226040533560...jpg Show all

Type here to search 28°C 2:05 PM 12/12/2022

The screenshot displays the Fournex Group portal interface. A modal window titled "Edit Slot B x 1 (71923)" is open, allowing for the configuration of a specific slot. The modal includes the following fields and options:

- Slot Enabled \***:  Yes
- Show Product Cost \***:  Yes
- Show Product Name \***:  Yes
- Show Product Image \***:  Yes
- Stock Limit Enabled \***:  No
- Stock List**: (Empty list)
- Product \***: BINGO - Chips Masala- 10 - 10
- Purchase Limit Per User\***: 0
- Purchase Limit Per Transaction \***: 0
- Purchase Limit Reset Per User Time \***: Every-Day
- Current Stock \***: 0

The background dashboard shows a grid of machine slots with their respective configurations and statuses. For example, "A x 4" is Disabled and Merged, "B x 4" is Disabled and Merged, "C x 1" is Enabled and Expanded, "C x 2" is Disabled and Merged, "C x 3" is Enabled and Expanded, and "C x 4" is Disabled and Merged. The portal footer indicates "Copyright © RIOTA Private Limited 2022" and "Version: 8.2.3".

fournex.riota.in/portal/transaction-list

Inbox - omguruco... நில் உரிமை (L... Inspector General o... Google WhatsApp Web iLovePDF | Online P... வட்டாரட்சியர் அ... Crop and Print Aad...

**FOURNEX Group** Welcome, Fournex Portal

**Transactions List**

Date Range ~ 12/12/2022 Search Type Machine Id Search Text X Sort By Time

Display Additional Info

Machine Id	Transactions Id	Client User Id	Client User Name	Employee Id	AuthType	Product	Cart Qty	Status	Amount
2VE0018420	22V2E1020112854121033	N/A	N/A	N/A	N/A (Disabled)	Paper Boat Aamras	1	SUCCESS	20
2VE0018340	22V2E1020112803141041	N/A	N/A	N/A	N/A (Disabled)	Bingo - Tomato & Chilli	1	SUCCESS	10
						Bingo - Tomato & Chilli	1	SUCCESS	10
						Mani mark seesam balls	1	SUCCESS	60
						Limca Pet 15	1	SUCCESS	15
						Limca Pet 15	1	DISPENSE FAILED	15
Fanta - 15	1	SUCCESS	15						
2VE0018444	22V2E10201128548414339	N/A	N/A	N/A	N/A (Disabled)	Mani mark coconut balls	1	SUCCESS	40
2VE0018454	22V2E10201128043514464	N/A	N/A	N/A	N/A (Disabled)	Lays spanish tomato 30	1	SUCCESS	30
Hersheys Copyright © RIOTA Private Limited 2022 Version: 8.2.3									

barath copy.docx barath copy.pdf 170226040533560...jpg Show all X

Type here to search 28°C 2:06 PM 12/12/2022

fournex.riota.in/portal/products

>Welcome, Fournex Portal

Machines List  
Current Stock  
Requests  
Machine Config  
Transactions  
Sales  
Refunds  
Transaction Config  
Products  
Brand  
Categories  
Sub Categories  
Product List  
Warehouse

Branch Level Client Level

Branch \* JP Trichy  
Branch Id: BR01017

Search Type Name Search Text X Sort By Created On

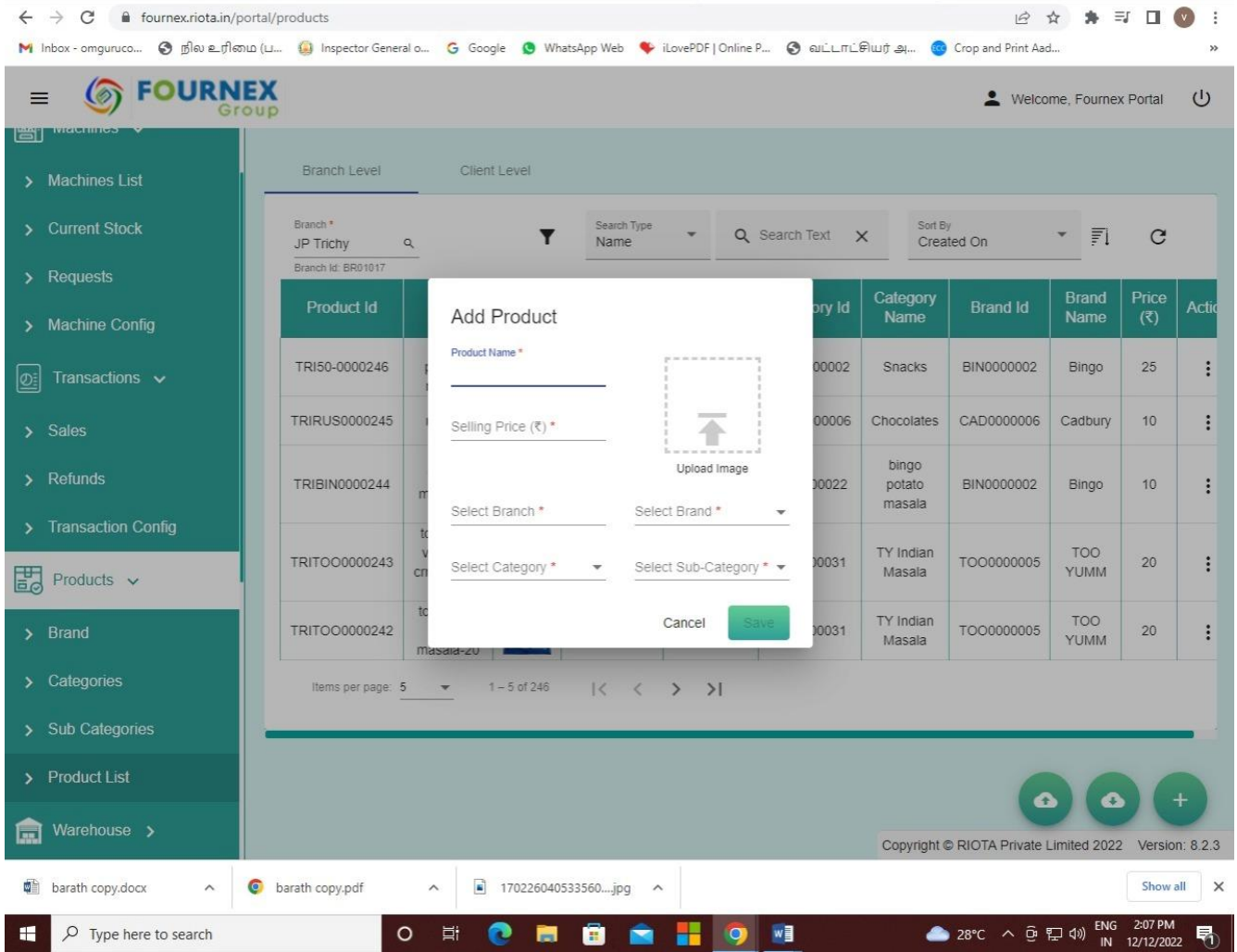
Product Id	Name	Image	Sub Category Id	Sub Category Name	Category Id	Category Name	Brand Id	Brand Name	Price (₹)	Action
TRI50-0000246	50-50-potazos mint-25		CHI0000002	Chips1	SNA0000002	Snacks	BIN0000002	Bingo	25	⋮
TRIRUS0000245	rusk-10		CHO0000006	Chocolates1	CHO0000006	Chocolates	CAD0000006	Cadbury	10	⋮
TRIBIN0000244	bingo tomoto madness-10		BIN0000023	bingo potato masala	BIN0000022	bingo potato masala	BIN0000002	Bingo	10	⋮
TRITOO0000243	tooyumm veg stick crm&onion-20		TYI0000032	TY Indian Masala	TYI0000031	TY Indian Masala	TOO0000005	TOO YUMM	20	⋮
TRITOO0000242	too yumm indian masala-20		TYI0000032	TY Indian Masala	TYI0000031	TY Indian Masala	TOO0000005	TOO YUMM	20	⋮

Items per page: 5 1 - 5 of 246

Copyright © RIOTA Private Limited 2022 Version: 8.2.3

barath copy.docx barath copy.pdf 170226040533560...jpg Show all X

Type here to search 28°C 2:06 PM 12/12/2022



## **CHAPTER VI**

### **CONCLUSION AND FUTURE ENHANCEMENT**

This machine is currently use to dispense snacks, medicines, napkin, toys. But along with technological advancement, we are planning to keep this in toxic atmosphere and non-habitat environment for human like, mines, undersea, Antarctic & Northern hemisphere. Also, this can be used to supply armors for soldiers. We are also planning to ship a machine to space to vending necessary supplies for human sustenance with a common currency across the globe. So that, any nationals can use the machine while visiting space or other planets.

Hence by this innovation, science and conscience of human can be taken to new habitat in our generation.

## APPENDIX

```
terraform {  
  required_providers {  
    aws = {  
      source = "hashicorp/aws"  
    }  
  }  
}
```

```
resource "aws_instance" "" {  
  ami      = var.ami  
  instance_type = var.instance_type
```

```
  network_interface {  
    network_interface_id = var.network_interface_id  
    device_index        = 0  
  }
```

```
  credit_specification {  
    cpu_credits = "unlimited"  
  }  
}
```

```
variable "network_interface_id" {  
  type = string  
  default = "network_id_from_aws"  
}
```

```
variable "ami" {  
  type = string  
  default = "ami-005e54dee72cc1d00"  
}
```

```
variable "instance_type" {  
  type = string  
  default = "t2.micro"  
}
```

```
resource "aws_db_instance" "default" {  
  allocated_storage = 10  
  db_name           = "mydb"  
  engine           = "mysql"  
  engine_version   = "5.7"  
  instance_class   = "db.t3.micro"  
  username         = "foo"  
  password         = "foobarbaz"  
  parameter_group_name = "default.mysql5.7"  
  skip_final_snapshot = true  
}
```

```
resource "aws_db_instance" "example" {  
  # ... other configuration ...  
  
  allocated_storage = 50  
  max_allocated_storage = 100  
}
```

```
resource "aws_db_instance" "db" {  
  s3_import {  
    source_engine       = "mysql"  
    source_engine_version = "5.6"  
    bucket_name        = "mybucket"  
    bucket_prefix      = "backups"  
    ingestion_role     = "arn:aws:iam::1234567890:role/role-xtrabackup-rds-restore"  
  }  
}
```



```
}  
  
provider "aws" {  
    access_key = "${var.aws_access_key}"  
    secret_key = "${var.aws_secret_key}"  
    region = "${var.region}"  
}  
  
module "s3" {  
    source = "<path-to-S3-folder>"  
    bucket_name = "your_bucket_name"  
}  
  
resource "aws_s3_bucket" "temps3" {  
    bucket = "${var.bucket_name}"  
    acl = "${var.acl_value}"  
}
```

```
variable "bucket_name" {}  
variable "acl_value" {  
    default = "private"  
}  
variable "aws_access_key" {  
    default = "<your_access_key>"  
}  
variable "aws_secret_key" {  
    default = "<your_secret_key>"  
}  
variable "region" {  
    default = "region"  
}
```

```
provider "aws" {  
    access_key = "${var.aws_access_key}"  
    secret_key = "${var.aws_secret_key}"  
    region = "${var.region}"  
}  
  
module "s3" {
```

```

source = "<path-to-S3-folder>"
bucket_name = "your_bucket_name"
}
resource "aws_s3_bucket" "temps3" {
  bucket = "${var.bucket_name}"
  acl = "${var.acl_value}"
}

```

```

variable "bucket_name" {}
variable "acl_value" {
  default = "private"
}
variable "aws_access_key" {
  default = "<your_access_key>"
}
variable "aws_secret_key" {
  default = "<your_secret_key>"
}
variable "region" {
  default = "region"
}

```

terraform init

terraform apply

```

aws ec2 run-instances --image-id ami-0e31b343ec333d397 --count 1 --instance-type t1.micro --key-name
sumanthPublicKey --security-groups-ids sg-0b56f3cdd4c96118a --region us-east-1

```

```

{
"Version": "2012-10-17",
"Statement": [
{
"Action": [
"autoscaling:PutLifecycleHook",

```

```
"autoscaling:DeleteLifecycleHook",
"autoscaling:RecordLifecycleActionHeartbeat",
"autoscaling:CompleteLifecycleAction",
"autoscaling:DescribeAutoscalingGroups",
"autoscaling:PutInstanceInStandby",
"autoscaling:PutInstanceInService",
"ec2:Describe*"
],
"Effect": "Allow",
"Resource": "*"
}
]
}

{
"Version": "2012-10-17",
"Statement": [
{
"Sid": "",
"Effect": "Allow",
"Principal": {
"Service": [
"codedeploy.us-east-1.amazonaws.com",
"codedeploy.us-west-2.amazonaws.com"
]
},
"Action": "sts:AssumeRole"
}
]
}

{
"Version": "2012-10-17",
"Statement": [
{
"Action": [
```

```

"s3:Get*",
"s3:List*"
],
"Effect": "Allow",
"Resource": "*"
}
]
}

```

```

ssh -i "nodepair.pem" PUBLIC_DNS_URLSudo susudo yum update
sudo yum install aws-cli
cd /home/ec2-user
aws s3 cp s3://aws-codedeploy-us-east-1/latest/install . --region us-east-1
chmod +x ./install
sudo ./install autoservice codedeploy-agent status

```

## REFERENCE

- [1] Android Studio 3.0 Development Essentials: Android 8 Edition by Neil Smyth in 2017
- [2] Embedded Programming with Android by Roger Ye in 2015
- [3] “Advanced Materials and Technologies for Micro/Nano-Devices, Sensors and Actuators” by Evgeni Gusev and Eric Garfunkel
- [4] “Wireless Sensor and Actuator Networks: Technologies, Analysis and Design” by Roberto Verdone Professor and Davide Dardari
- [5] “Mobile Point Sensors and Actuators in the Controllability Theory of Partial Differential Equations” by Alexander Y Khapalov
- [6] AWS Serverless Computing: A beginner's guide to using AWS Lambda, Amazon API Gateway, and services from Amazon Web Services 1st Edition, Kindle Edition by Scott Patterson