

**COMSATS University Islamabad**

**Abbottabad, Pakistan**

**Student Information System**

***By***

**BASIT IQBAL CIIT/FA2021-BSE-050/ATD**

**FATIMA AFTAB CIIT/FA2021-BSE-088/ATD**

**FAWAD IQBAL CIIT/FA2021-BSE-012/ATD**

**FAIZAN CIIT/FA2021-BSE-011/ATD**

***Supervisor*Sana Malik**

***Bachelor of Science in Computer Science (2023-2023)***

**The candidate confirms that the work submitted is their own and appropriate  
 credit has been given where reference has been made to the work of others**.

A blue and white logo

Description automatically generated

**EXECUTIVE SUMMARY**

The Student Information System is a solution that helps schools and colleges manage student information more efficiently and encourages better teamwork. It replaces manual systems that often lead to mistakes and make it hard to make smart decisions. This system uses a database to store student details like their personal information, grades, and more. It ensures that everyone has access to accurate and up-to-date information, which helps avoid mistakes and promotes transparency.

With this system, students can easily access their records. Administrators can generate reports, analyze data, and make better decisions.

Implementing the Student Information System brings many benefits, including smoother data management, less administrative work, improved teamwork, and better decision-making. Institues can use accurate information to help students succeed, provide the right support, and make the most of their resources.

In conclusion, the Student Information System is a helpful solution for schools and colleges to manage student information and promote collaboration. By using technology to streamline data management, increase transparency, and encourage teamwork, this system contributes to better educational outcomes and a positive environment for students, teachers, and administrators.

Table of Contents

[**1.** **Introduction** 7](#_Toc140108736)

[**1.1** **Relevance to Course Modules** 7](#_Toc140108737)

[**1.2** **Project Background** 7](#_Toc140108738)

[**1.3** **Literature Review** 7](#_Toc140108739)

[**2.** **Problem Definition** 8](#_Toc140108740)

[**2.1** **Problem Statement** 8](#_Toc140108741)

[**2.2** **Proposal** 8](#_Toc140108742)

[**3.** **ERD Diagram** 9](#_Toc140108743)

[**3.1** **Level 0 ERD Diagram** 9](#_Toc140108744)

[3.1.1 Description: 9](#_Toc140108745)

[3.1.2 Entities: 9](#_Toc140108746)

[3.1.3 Diagram: 10](#_Toc140108747)

[**3.2** **Level 1 ERD Diagram** 11](#_Toc140108748)

[3.2.1 Relations: 11](#_Toc140108749)

[3.2.2 Diagram: 11](#_Toc140108750)

[**3.3** **Level 2 ERD Diagram** 12](#_Toc140108751)

[3.3.1 Resolve Many to Many Relations Of ERD: 12](#_Toc140108752)

[3.3.2 Diagram 12](#_Toc140108753)

[**3.4** **ERD Diagram After Normalization** 13](#_Toc140108754)

[3.4.1 Description: 13](#_Toc140108755)

[3.4.1 Diagram: 13](#_Toc140108756)

[**4.** **Normalization** 14](#_Toc140108757)

[**4.1.** **First Normal Form (1NF)** 14](#_Toc140108758)

[**4.2.** **Second Normal Form (2NF)** 14](#_Toc140108759)

[**4.3.** **Third Normal Form (3NF)** 15](#_Toc140108760)

[**4.4.** **Tables after normalization** 15](#_Toc140108761)

[4.4.1. fruitList: 15](#_Toc140108762)

[4.4.2. fruitList\_item: 15](#_Toc140108763)

[4.4.3. Items: 16](#_Toc140108764)

[4.4.4. Veg\_List: 16](#_Toc140108765)

[4.4.5. VegList\_item: 16](#_Toc140108766)

[4.4.6. Admin: 16](#_Toc140108767)

[4.4.7. AdminContact: 17](#_Toc140108768)

[4.4.8. Customers: 17](#_Toc140108769)

[4.4.9. CustomerContact: 17](#_Toc140108770)

[4.4.10. Vendor: 17](#_Toc140108771)

[4.4.11. VendorContact: 17](#_Toc140108772)

[**5.** **Translation Schema:** 18](#_Toc140108773)

[**6.** **Conclusion** 24](#_Toc140108774)

# **Introduction**

The Student Information System is a solution that helps schools and colleges manage student information more efficiently and encourages better teamwork. It replaces manual systems that often lead to mistakes and make it hard to make smart decisions. This system uses a database to store student details like their personal information, grades, and more. It ensures that everyone has access to accurate and up-to-date information, which helps avoid mistakes and promotes transparency.

## **Relevance to Course Modules**

## The project is implemented by using the different concepts of Data Structure and Algorithms. This project is implemented by using a singly link list. It is used to store the data of students in institute. By using link list, it can store data, delete data, and search or find data.

* 1. **Project Background**

The Student Information System is a solution that helps schools and colleges manage student information more efficiently and encourages better teamwork. It replaces manual systems that often lead to mistakes and make it hard to make smart decisions. This system uses a database to store student details like their personal information, grades, and more. It ensures that everyone has access to accurate and up-to-date information, which helps avoid mistakes and promotes transparency.

Implementing the Student Information System brings many benefits, including smoother data management, less administrative work, improved teamwork, and better decision-making. Institues can use accurate information to help students succeed, provide the right support, and make the most of their resources.

## **Literature Review**

## There are a lot of projects on the student information system. Such as,

* Classe365
* Creatrix Campus
* SIS APP
* SIS Punjab

# **Proposal**

We are proposing the implementation of a modern Student Information System (SIS) for our organization. The objective of this proposal is to improve the management of student data, streamline administrative tasks, and enhance communication among stakeholders. The proposed SIS will automate processes such as enrollment, registration, and academic records management. It will also include modules for students and administrators to access and update relevant information. By implementing the SIS, we aim to increase administrative efficiency, ensure data accuracy, and improve communication within our educational institution. We seek support and approval for this initiative, believing it will bring long-term benefits and keep us at the forefront of educational technology.

# **Functional Requirements of System**

## **Admin’s Requirements**

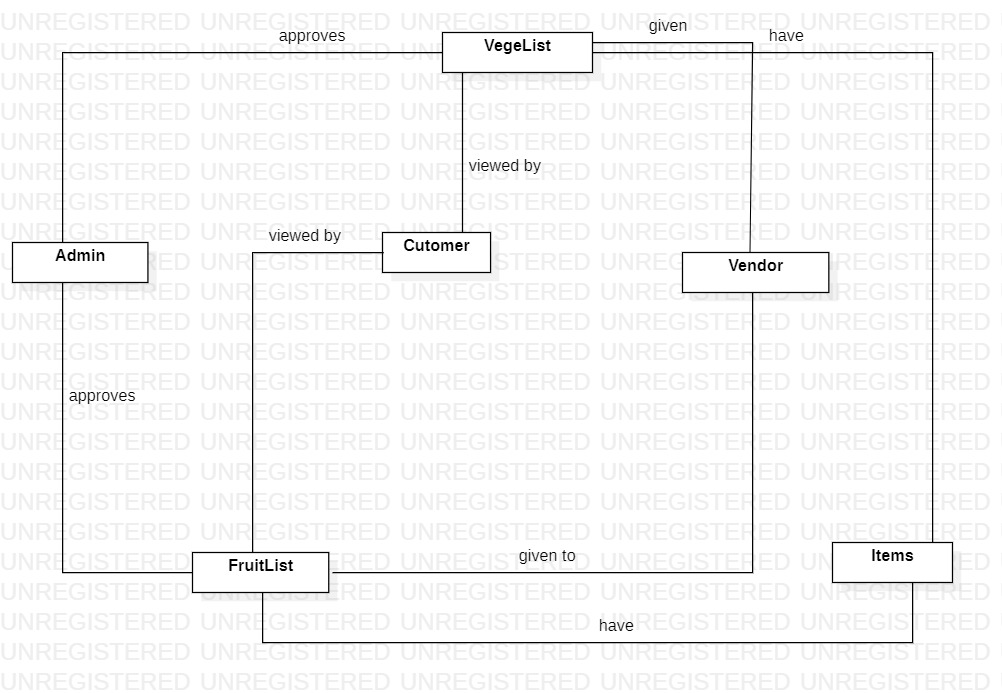
### Login:

Admin can login to the system by entering his name and password. If the details are correct, then admin can view his dashboard. Otherwise, the error message will display.

### Logout:

Admin can logout to the system if he clicks on logout button.

### 3.1.3 :

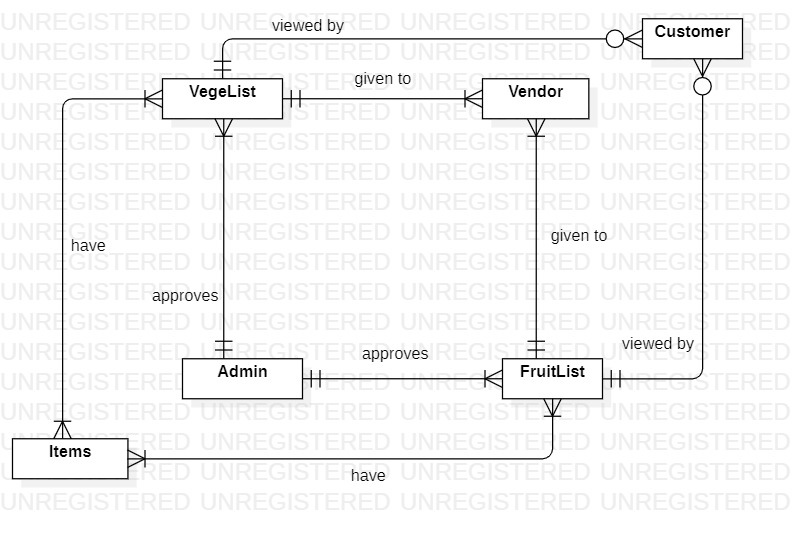


## **Level 1 ERD Diagram**

### 3.2.1 Relations:

* Admin can approve one or many list of fruits and vegetables , at least one list of fruit and vegetable can be approved by admin daily.
* Fruits and Vegetables list can be viewed by many customers or may not be viewed by a single customer daily.
* Fruits and Vegetables list can be given to one or many vendors, one vendor gets only one list.
* Fruits and Vegetables lists have at least one or many items, but items can be present in many lists or may not be present in any list.

### 3.2.2 Diagram:

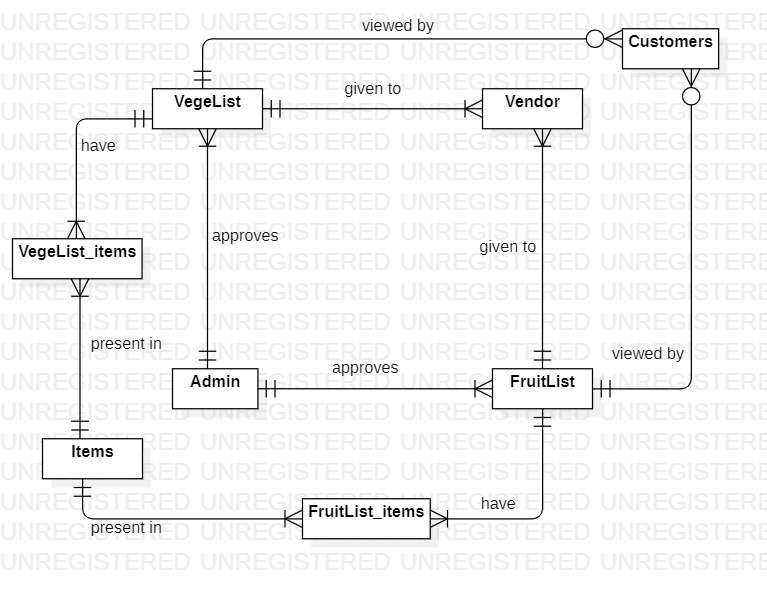


## **Level 2 ERD Diagram**

### 3.3.1 Resolve Many to Many Relations Of ERD:

The above description is level 1 of the sabzi management system, It contains many to many relations between entities so to solve those many to many relations we will draw level 2 by entering the gerund entities between those entities which have many to many relations. We will just insert the gerund entity between those entities which have many to many relations and then we will put the primary keys of both the entities into the gerund entity as foreign key .

### 3.3.2 Diagram

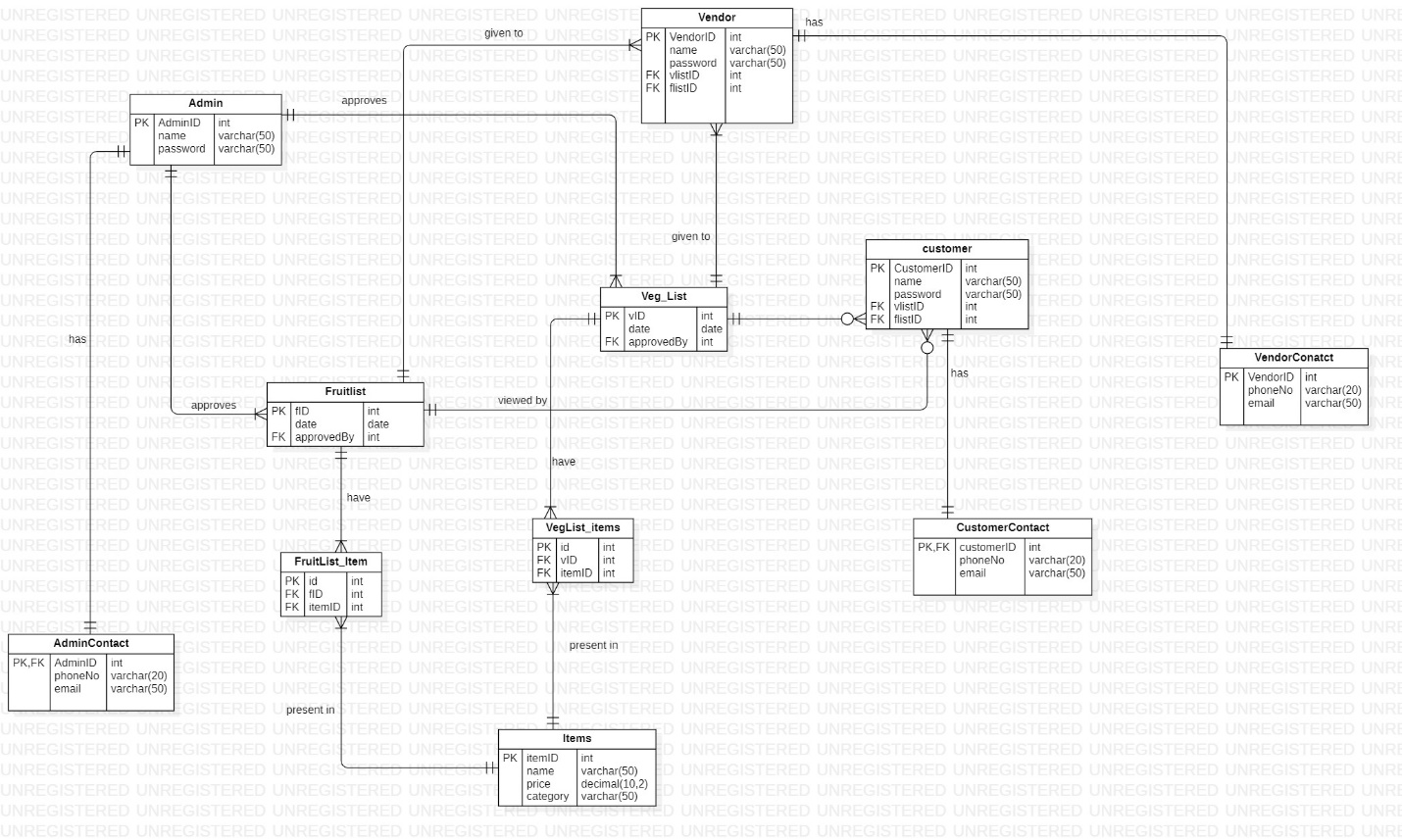


## **ERD Diagram After Normalization**

### 3.4.1 Description:

This is the final version of ER diagram that was made after normalization, the need that was felt to make it was that the number of entities increased after normalization.

### Diagram:



# **Normalization**

There are seven tables.

1. Admin( AdminID, name, password, phoneNo, e-mail)
2. Customer( CustomerID, name, password, e-mail, phoneNo, vlistID, flistID)
3. Vendor ( VendorID, name, password, phoneNo, email, vlistID, flistID)
4. Items ( itemID, name, price, category)
5. Fruitlsit (fID, date, approvedby)
6. Veg\_list(vID, date, approvedby)
7. Fruitlist\_item( id, fID, itemID)
8. Veglist\_item( id, vID, itemID)

## **First Normal Form (1NF)**

As the tables are in first normal form because there are no repeating groups in them.

## **Second Normal Form (2NF)**

The tables which are already in 2NF are ,

1. Items ( itemID, name, price, category)
2. Fruitlsit (fID, date, approvedby)
3. Veg\_list(vID, date, approvedby)
4. Fruitlist\_item( id, fID, itemID)
5. Veglist\_item( id, vID, itemID)

Now tables which are not in 2NF are,

1. Admin( AdminID, name, password, phoneNo, e-mail)
2. Customer( CustomerID, name, password, e-mail, phoneNo, vlistID, flistID)
3. Vendor ( VendorID, name, password, phoneNo, email, vlistID, flistID)

The reason behind not being 2NF is that in all three tables the attribute's phoneNo and e-mail are not dependent upon admin ID, customer ID, and vendor ID respectively.

So, after resolving the to 2NF tables would divide as,

1. Admin( AdminID, name, password)
2. Customer( CustomerID, name, password, vlistID, flistID)
3. Vendor ( VendorID, name, password, vlistID, flistID)
4. AdminContact( AdminID, phoneNo, email)
5. CustomerContact(CustomerID, phoneNo, email)
6. VendorContact(VendorID, phoneNo, email)

## **Third Normal Form (3NF)**

All the tables are in 3NF as there is only one primary key as PK in each table.

## **Tables after normalization**

### fruitList:

|  |  |  |
| --- | --- | --- |
| fID | Date | approvedBy |
| 21 | 2023-07-1 | 1 |
| 22 | 2023-07-2 | 2 |
| 23 | 2023-07-3 | 1 |
| 24 | 2023-07-4 | 1 |

### fruitList\_item:

|  |  |  |
| --- | --- | --- |
| ID | fID | itemID |
| 6 | 21 | 1 |
| 7 | 22 | 2 |
| 8 | 23 | 3 |
| 9 | 24 | 4 |

### Items:

|  |  |  |  |
| --- | --- | --- | --- |
| ItemID | name | price | category |
| 1 | Apple | 1.99 | Fruit |
| 2 | Banna | 0.99 | Fruit |
| 3 | Carrot | 0.49 | Vegetable |
| 4 | Orange | 1.49 | Fruit |

### Veg\_List:

|  |  |  |
| --- | --- | --- |
| vID | date | approvedBy |
| 11 | 2023-07-1 | 1 |
| 12 | 2023-07-2 | 2 |
| 13 | 2023-07-3 | 1 |
| 14 | 2023-07-4 | 2 |

### VegList\_item:

|  |  |  |
| --- | --- | --- |
| ID | vID | itemID |
| 5 | 11 | 3 |
| 6 | 12 | 3 |
| 7 | 13 | 3 |
| 9 | 24 | 4 |

### Admin:

|  |  |  |
| --- | --- | --- |
| AdminID | name | password |
| 1 | Basit Iqbal | 123 |
| 2 | Fatima Aftab | 456 |

### AdminContact:

|  |  |  |
| --- | --- | --- |
| AdminID | phoneNo | e-mail |
| 1 | 03122456789 | [basitiqbal@yahoo.com](mailto:basitiqbal@yahoo.com) |
| 2 | 03152345789 | vasi@gmail.com |

### Customers:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CustomerID | name | password | vlistID | flistID |
| 1 | Ahsan | 050 | 4 | 4 |
| 2 | Basit | 012 | 5 | 5 |
| 3 | Fatima | 456 | 6 | 6 |

### CustomerContact:

|  |  |  |
| --- | --- | --- |
| CustomerID | phoneNo | e-mail |
| 1 | 03122456789 | [aliqbal@yahoo.com](mailto:aliqbal@yahoo.com) |
| 2 | 03152345789 | wasimsi@gmail.com |

### Vendor:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| VendorID | name | password | vlistID | flistID |
| 1 | Aysha | 050 | 1 | 1 |
| 2 | Noor | 012 | 3 | 3 |

### VendorContact:

|  |  |  |
| --- | --- | --- |
| VendorID | phoneNo | e-mail |
| 1 | 03122499789 | [fawad@yahoo.com](mailto:fawad@yahoo.com) |
| 2 | 03152347789 | ali@gmail.com |

# **Translation Schema:**

**1-Table Name:** Admin

**Primary key:** AdminID

**Description:** This table used to store the details of all Admins

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Description** |
| AdminID | INT |  | Admin no |
| name | Varchar | 50 | Name of Admin |
| password | Varchar | 50 | Password of Admin |

**2-Table Name:** AdminContact

**Primary key:** AdminId.

**Foreign key:** AdminId, References Admin (AdminId)

**Description:** This table contains information of Admins contacts

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Description** |
| AdminID | INT |  | Admin id |
| phoneNo | varchar | 20 | Phone number of admin |
| email | Varchar | 50 | Email of admin |

**3-Table Name:** Customer

**Primary key:** CustomerID

**Foreign** **key**: vlistID: References Veg\_List (vID),

flistID: References fruitList ( fID).

**Description:** This table used to store the names of all Customers

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Description** |
| CustumerID | INT |  | Admin no |
| name | Varchar | 50 | Name of Customer |
| password | Varchar | 50 | Password of Customer |
| vlistID | INT |  | Veg\_List id |
| flistID | INT |  | FruitList id |

**4-Table Name:** CustomerContact

**Primary key:** CustomerID.

**Foreign key:** CustomerID, References Costumer (CustomerID)

**Description:** This table contains information of Customer contacts

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Description** |
| CustomerID | INT |  | Admin id |
| phoneNo | varchar | 20 | Phone number of Customer |
| email | Varchar | 50 | Email of Customer |

**5-Table Name:** Vendor

**Primary key:** VendorID

**Foreign** **key**: VlistID: References Veg\_List (vID),

flistID: References fruit List ( fID).

**Description:** This table used to store the names of all Vendors

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Description** |
| VendorID | INT |  | Vendor no |
| name | Varchar | 50 | Name of Vendor |
| password | Varchar | 50 | Password of Vendor |
| vlistID | INT |  | Veg\_List id |
| flistID | INT |  | fruitList id |

**6-Table Name:** VendorContact

**Primary key:** VendorID

**Foreign key:** VendorID, References Vendor (VendorID)

**Description:** This table contains information of Vendors contacts

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Description** |
| VendorID | INT |  | Vendor id |
| PhoneNo | varchar | 20 | Phone number of Vendor |
| email | Varchar | 50 | Email of Vendor |

**7-Table Name:** veg\_List

**Primary key:** vID.

Foreign key: approvedBy , References Admin(AdminID)

**Description:** This table contains information of Admins contacts

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Description** |
| vID | INT |  | Veg\_List id |
| date | date |  | Date of issue |
| approvedBy | INT |  | Admin id |

**8-Table Name:** fruitList

**Primary key:** fID.

**Foreign** **key**: approvedBy , References Admin(AdminID)

**Description:** This table contains information of Admins contacts

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Description** |
| fID | INT |  | fruitList id |
| date | date |  | Date of issue |
| approvedBy | INT |  | Admin id |

**9-Table Name:** items

**Primary key:** itemID.

**Description:** This table contains information of Admins contacts

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Description** |
| itemID | INT |  | item id |
| name | Varchar | 50 | Name of item |
| price | INT |  | Price of a particular item |
| category | varchar | 50 | Fruit or vegetable |

**10-Table Name:** fruitlist\_items

**Primary key:** ID.

**Foreign key:** fID, References fruitlist(fID),

itemID References item(itemID).

**Description:** This table is a gerund entity that is made to solve the many to many relationships.

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Description** |
| id | INT |  | item id |
| fID | INT |  | Fruitlist id |
| itemID | INT |  | Item id |

**11-Table Name:** veglist\_items

**Primary key:** ID.

**Foreign key:** vID, References veg\_list(vID),

itemID References item(itemID).

**Description:** This table is a gerund entity that is made to solve the many to many relationships.

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Description** |
| id | INT |  | item id |
| vID | INT |  | Veg\_list id |
| itemID | INT |  | Item id |

# **Conclusion**

In conclusion, the proposed Sabzi Mandi database system presents a transformative solution to the longstanding challenges faced in Abbottabad's vegetable market. By leveraging advanced technology, this comprehensive system enables seamless collaboration, transparency, and accuracy among customers, vendors, and supervisors. The elimination of discrepancies and unauthorized modifications to the approved list will combat price inflation and promote fair market practices. With efficient data storage and analytics capabilities, the government can make informed decisions to drive the growth and efficiency of the Sabzi Mandi, benefiting the local economy and all stakeholders involved.