

SCHOOL OF COMPUTER SCIENCES

UNIVERSITI SAINS MALAYSIA

CMT221 : Database Organization and Design

Semester I, Academic Session : 2024/2025

Report 2 - System Implementation

Group 12

LECTURER’S Name: Dr. Hazrina Yusof Hamdani

Plant Nursery Inventory Management System by YnB

| Names | Matric No | USM Email Addresses | Module |
| --- | --- | --- | --- |
| Shivabala A/L Ganeish **(Leader)** | 22304163 | shivabalaganeish@student.usm.my | 2 |
| Queek Yu Feng | 22301758 | queekyufeng@student.usm.my | 1 |
| Pavithran A/L Muthiah | 22303521 | pavithran040315@student.usm.my | 4 |
| Timothy Thomas | 22302352 | mothy2180@student.usm.my | 3 |

Date of submission

19 January 2025

### Table of Contents

| **Contents** | **Page** |
| --- | --- |
| **Cover Page** | 1 |
| **Table of Contents** | 2 |
| **1.0 Introduction, Requirements, Business rules and Entity relationship modelling**  1.1 Introduction  1.1.1 Description of the Organization  1.1.2 Current System and its Problem  1.1.3 Motivation of proposed project  1.1.4 How the Proposed Database System will benefit the Organization for Each  Module  1.1.4.1 Inventory Management Module  1.1.4.2 Biodiversity Tracking Module  1.1.4.3 Waste Reduction & Sustainability Module  1.1.4.4 Environmental Impact Module  1.2 Requirements  1.2.1 Inventory Management Module  1.2.2 Biodiversity Tracking Module  1.2.3 Waste Reduction & Sustainability Module  1.2.4 Environmental Impact Module  1.3 Business rules  1.3.1 Inventory Management Module  1.3.2 Biodiversity Tracking Module  1.3.3 Waste Reduction & Sustainability Module  1.3.4 Environmental Impact Module  1.4 Entity relationship modelling  1.4.1 Inventory Management Module  1.4.2 Biodiversity Tracking Module  1.4.3 Waste Reduction & Sustainability Module  1.4.4 Environmental Impact Module  1.4.5 Overall ERD | 4  4  4  5  6  6  7  8  8  9  9  10  10  11  12  12  12  12  12  13  13  16  19  21  24 |
| **2.0 Normalization**  2.1 Inventory Management Module  2.2 Biodiversity Tracking Module  2.3 Waste Reduction & Sustainability Module  2.4 Environmental Impact Module | 25  25  26  27  29 |
| **3.0 Data Dictionary**  3.1 Inventory Management Module  3.2 Biodiversity Tracking Module  3.3 Waste Reduction & Sustainability Module  3.4 Environmental Impact Module | 30  30  30  30  30 |
| **4.0 Database implementation**  4.1 Inventory Management Module  4.2 Biodiversity Tracking Module  4.3 Waste Reduction & Sustainability Module  4.4 Environmental Impact Module | 31  31  31  31  31 |
| **5.0 Front-end system design and implementation**  5.1 Inventory Management Module  5.2 Biodiversity Tracking Module  5.3 Waste Reduction & Sustainability Module  5.4 Environmental Impact Module | 31  32  32  33  34 |
| **6.0 Project Problems and Pitfalls** | 34 |
| **7.0 Conclusions** | 35 |
| **8.0 References** | 36 |
| **9.0 Appendix** | 38 |

### 

**1.0 Introduction, Requirements, Business rules and Entity relationship modelling**

#### 1.1 Introduction

**1.1.1 Description of the Organization**

Reducing waste while being economically sound and ensuring a minimal impact on the environment is what Sustainable Development Goal 12 of the United Nations is aimed for and it is addressed through the promotion of responsible consumption and production. Minimizing waste associated with food and any other materials, as well as implementing policies encouraging recycling and optimizing the use of resources fall under the umbrella of sustainable practices. Some of the targets to be achieved include a 50 percent reduction in global food waste, sustainable public procurement, and the environmental-friendly approach of production aims to be integrated within businesses by 2030. YnB Company is one such business that ensures that its operations are aligned with SDG 12. Providing eco-friendly gardening supplies, aiming to conserve biodiversity, cutting down on waste, and making the customers aware of sustainable options are some of the measures taken by the company. Therefore, through these initiatives, YnB works towards global factors such as climate change, resource depletion, ecosystem conservation, and sustainable economic and social development. In conclusion, YnB is actively pursuing SDG 12 to endorse a safer environment for all now and in the future.

#### 1.1.2 Current System and its Problems

YnB now uses spreadsheets and paper records as the main tools for its manual inventory and operational data management system. Although the company's fundamental needs have been satisfied by these conventional approaches, they are becoming less and less suitable for handling the rising complexity of YnB's operations and sustainability objectives. The company's efficiency and environmental goals are hampered by a number of serious issues brought about by this manual technique.

Firstly is inaccurate inventory tracking, as manual updates often lead to inconsistencies and outdated stock records. This leads to frequent mistakes that interfere with sales and restocking procedures, like missing items, inaccurate stock counts, and inconsistencies.These errors also make it difficult to predict demand, which results in shortages of some products and overstocking of others. Plants that are overstocked frequently go unsold and end up as garbage, which will contribute to financial loss and environmental harm.

Operational inefficiency is another serious issue. Updating records, cross-checking data, and retrieving data for decision-making all take a lot of time and effort using the manual approach. Repetitive duties like manually updating stock levels, monitoring for low inventory, or producing reports frequently need staff personnel to spend significant time. Overall productivity is impacted by these inefficiencies, which slow down vital operations like sales, purchasing, and customer support.

The issue is made worse by the lack of integrated systems. Data on biodiversity and the environment cannot be tracked in real time with YnB's current configuration. The business doesn't have a methodical approach to tracking recycling, trash production, and other sustainability indicators. This disparity hinders YnB's capacity to pinpoint problem areas, establish quantifiable objectives, and illustrate its advancements in environmental sustainability.

Lastly, YnB is unable to efficiently monitor its carbon footprint or resource usage due to a lack of instruments for assessing the environmental impact of operations. The company is unable to integrate its operations with its sustainability purpose or the United Nations SDG 12: Responsible Consumption and Production since crucial activities including energy usage, water consumption, and greenhouse gas emissions are not tracked.

A strong, integrated database solution that can resolve these inefficiencies, optimize operations, and uphold YnB's dedication to environmental stewardship is desperately needed, as seen by the shortcomings of the current system.

#### 1.1.3 Motivation for the Proposed Project

The proposed project seeks to address these challenges by developing an integrated database system that automates and streamlines key processes. This system is motivated by the need to enhance YnB's operational efficiency, reduce waste, and ensure alignment with sustainability objectives. By implementing a robust database solution, YnB can transition from a fragmented and error-prone system to one that provides real-time insights, fosters informed decision-making, and actively supports its environmental mission.

The motivation behind the Plant Nursery Inventory Management System stems from the need to address key inefficiencies and sustainability challenges faced by a plant nursery dedicated to environmental stewardship and responsible consumption. Currently, plant nurseries rely on manual methods, such as spreadsheets and paper records, for inventory management. This outdated approach is time-consuming, prone to human error, and lacks real-time data visibility, leading to operational inefficiencies in stock replenishment, sales processes, and overall customer satisfaction.

Additionally, they struggle with waste management and sustainability tracking. Without proper tools for monitoring inventory trends or forecasting demand, the nursery often faces overstocking and plant waste, which contradicts its commitment to environmental sustainability. The absence of systems to measure environmental metrics, such as carbon footprint, energy consumption, and biodiversity preservation, further hinders a plant nursery’s ability to align its practices with Sustainable Development Goal (SDG) 12: Responsible Consumption and Production.

Therefore, YnB proposes a solution, as we specialise in biodiversity preservation to solve the lack of a reliable system to track the growth, origins, or conservation status of native and endangered plant species. This limits the nursery’s ability to fully support its mission of biodiversity enhancement and ecological responsibility. The proposed solution seeks to bridge these gaps by providing a robust, digital solution that ensures real-time inventory tracking, minimizes waste, and monitors biodiversity and environmental impact.

This project is more than a technological upgrade; it is a strategic tool to help plant nurseries achieve operational efficiency, sustainability leadership, and data-driven decision-making. Through this project, YnB will assist nurseries to streamline their processes, reduce their environmental footprint and establish themselves as a model for sustainable business practices within the nursery industry.

#### 

#### 1.1.4 How the Proposed Database System Will Benefit the Organization

##### 1.1.4.1 Inventory Management Module

#### The proposed inventory management module will significantly benefit YnB Company by providing real-time tracking of plants and supplies. Employees and management will have access to up-to-date inventory records, including stock levels, storage locations, and last updated timestamps, which eliminates errors caused by manual tracking and enhances decision-making. The automation of inventory updates will streamline operations, reduce administrative workloads, and allow employees to focus on more critical tasks. Furthermore, the system facilitates efficient location of inventory items within storage areas, improving operational productivity.

#### By ensuring accurate inventory tracking, the module will help reduce waste caused by overstocking and prevent the loss of plants and supplies due to spoilage. Integration with the waste management system will allow the organization to monitor and manage waste effectively, supporting its sustainability goals. Additionally, the module enhances customer service by ensuring that orders are fulfilled promptly without delays caused by stockouts. Employees can also easily track which plants are included in specific orders, enabling efficient order preparation.

#### The module will also play a crucial role in achieving YnB’s sustainability objectives by optimizing resource allocation and monitoring the effectiveness of supplies used for plant care. It enables detailed reporting and analysis of inventory trends, such as frequently purchased plants or commonly used supplies, aiding in demand forecasting and stock level optimization. This ensures resources are allocated efficiently and reduces unnecessary purchases.

#### Moreover, the module is scalable and flexible and the inventory management module can accommodate YnB’s growth, allowing the addition of new plants, supplies, and storage locations without compromising performance. It also supports future integration with other modules, such as customer and order management, ensuring seamless operations across the organization. Overall, the module will improve efficiency, reduce waste, and align YnB’s operations with its environmental mission, creating a robust foundation for sustainable growth.

##### 1.1.4.2 Biodiversity Tracking Module

#### The Biodiversity Tracking Module will significantly benefit the plant nurseries by providing a structured and efficient way to manage comprehensive records of plant varieties, particularly those that are native, rare, or endangered. Utilizing the module, they can systematically document crucial data such as plant species, growth records, and locations, as illustrated in the entity-relationship diagram (ERD). This database structure ensures detailed tracking of plants' health, growth progress, and their environmental conditions, which are pivotal for biodiversity conservation.

The module's capability to link plant records to specific locations and growth logs allows plant nurseries to monitor plants' contributions to conservation efforts effectively. For instance, through the "Plant\_Growth\_Record" and "Growth\_Log" entities, the system tracks growth metrics like height, health conditions, and stem circumference, enabling them to make informed decisions on plant care and maintenance. The "Plant\_Location" entity further enhances this functionality by associating plants with specific environmental types, fostering better management of ecological conditions.

By systematically managing this data, the module empowers plant nurseries to align their operations with their biodiversity conservation goals. Furthermore, the comprehensive database can serve as an educational tool, supporting initiatives to raise environmental awareness and promote sustainable practices. This integration of data management and environmental education positions YnB as a leader in biodiversity preservation and ecological responsibility

##### 1.1.4.3 Waste Reduction & Sustainability Module

#### The Waste Reduction & Sustainability Module is a vital component of the proposed database system, designed to help YnB minimize waste while enhancing its commitment to environmental stewardship. By enabling waste disposal methods and recycling efforts, this module ensures that waste is effectively managed and sustainable practices are well-documented. With real-time data insights, YnB can identify inefficiencies, reduce overstocking, and adopt proactive measures to minimize the environmental impact of its operations.

#### In addition to waste management, the module will generate detailed reports that track progress toward waste reduction targets and highlight areas for improvement. These reports will not only support internal decision-making but also provide transparency to stakeholders about YnB's environmental initiatives. By offering actionable insights, the system will help the company optimize resource usage, such as energy and water, and promote eco-friendly alternatives. This aligns with YnB’s mission of integrating sustainability into all aspects of its operations.

#### Furthermore, the module supports YnB’s alignment with the United Nations SDG 12 (Responsible Consumption and Production) by encouraging sustainable consumption practices and resource optimization. By embedding sustainability metrics into daily operations, YnB can systematically reduce its carbon footprint and demonstrate measurable progress toward global sustainability goals. Ultimately, this module will enable YnB to uphold its environmental mission while achieving operational excellence and fostering long-term growth.

##### 1.1.4.4 Environmental Impact Module

#### The Environmental Impact Module will provide insights into YnB’s carbon footprint and resource usage. By analyzing data on activities such as fertilization, energy consumption, and transportation, this module will help YnB monitor and reduce its environmental impact. It will generate reports on energy usage, water consumption, and greenhouse gas emissions, allowing the company to set and achieve data-driven sustainability goals.The Environmental Impact Module database system will significantly enhance the YnB organisation’s ability to manage plant operations, resource management, and environmental impact measures.

#### By centralizing data on plants, activities, resources, and environmental impacts, the system ensures provided data management and ease access to critical information. This establishes clear relationship’s types between entities which enables efficient tracking of activities, resources, and their environmental implications. The analysis report generation provides valuable insights into resource utilization and environmental implication measures which aid in data-driven decision-making.

Resource and waste management are the main benefits of the database system. The detailed tracking of resource usage in activities for plants in the nursery allows the YnB organization to identify flaws and optimize utilization, reducing wastages and costs. The waste management component ensures proper documentation of waste type, location, quantity, and aligning operations with sustainability and regulatory requirements. Additionally, environmental implications such as emissions and thresholds are monitored to proactively mitigate risks and maintain compliance with environmental standards.

This system supports the YnB organization’s commitment to sustainable development goals (SDG). The database can evolve with the YnB’s needs, making it a valuable asset for achieving long-term operational efficiency and sustainability goals.

### 1.2 User Requirements

### 1.2.1 Inventory Management Module

* Admin can register to an account by entering information like name, phone number, email address, address and date joined.
* Admin should have the ability to track each plant within the nursery's inventory, including details such as the plant's name, category, available stock quantity, selling price, the most recent update time and species .
* Once a customer is registered, their information, including name, email, phone number, and address, can be updated as needed. However, customer records cannot be deleted to ensure historical data integrity.
* Customers are allowed to place multiple orders over time, but each order must be associated with exactly one customer. Customers can track their orders, including the order date, total amount, and current status.
* Admin can review daily inventory transaction records for plants, including details about the order date, total amount of order and order status. They can access and track detailed information about the plants included in each order, such as their name, category, price and species.
* Admin should be able to track every plant for the activities carried out to maintain the plant's health. They are able to track which activities are carried out with its description of it.
* Admin are able to track the plant inside the plant record and also record the plant's planting date.

#### 1.2.2 Biodiversity Tracking Module

* Admin should be able to store detailed information about each plant species, including its scientific classification (e.g., genus, species, family), common name, and conservation status (e.g., native, rare, endangered).
* Admin must be able to log new plant varieties into the system, ensuring accurate records of all species managed by the nursery.
* Admin can track growth metrics for plants, such as height, stem circumference, and health condition, and update these metrics regularly to monitor development.
* Endangered plant species should be highlighted within the system for easy identification and prioritization of monitoring and care.
* Admin should be able to generate biodiversity reports summarizing metrics such as the number of endangered species, growth trends, and the nursery's overall contribution to biodiversity conservation.
* Admin should have the ability to update plant records, including modifications to growth metrics, conservation status, and other key details.
* The system must allow data sharing with conservation organizations, enabling nurseries to collaborate and contribute to larger biodiversity preservation initiatives.
* Admin should be able to monitor how plants align with biodiversity goals and ensure compliance with conservation requirements.

#### 1.2.3 Waste Reduction & Sustainability Module

* Admin should be able to create and manage waste records, including waste type, quantity, waste date, and related resources.
* Admin should manage recycling center details, including name, location, capacity, and link waste records to appropriate centers.
* Admin should track resources used, including name, cost, quantity used, and remaining stock, and link them to waste records.
* Admin should define sustainability goals with descriptions, deadlines, and progress linked to specific waste records.
* Admin should set and track waste reduction goals, linking them to waste records to measure achievement.
* Admin should have real-time data visibility to identify inefficiencies and optimize waste management.

#### 

#### 1.2.4 Environmental Impact Module

* Admin should be able to manage details of plants including their name, category, stock quantity, price and the last updated time and each plant should be linked to its respective species, order, storage location and respective analysis report.
* Admin should be able to record and manage activities performed on each plant including activity name, and description such as fertilization, water consumption and carbon emission and it must be linked to plants and their corresponding environmental implication.
* Admin should be able to track how much and what type of resource such as water, electricity and fertilizer are used in each activity.
* Admin should calculate and store environmental implications caused by activities such as emission or resource usage with respective information about the environmental impact type, measurement, and threshold and monitoring period.
* Admin should be able to generate analysis reports summarizing the environmental impact data for each plant location including details like report date, summary of findings, and should be linked to environmental implications and specific plant locations.
* The admin should record details of waste produced during the resource usage in activity for each plant including the information about the waste type, quantity, date of waste generation, and location of plant.
* Admin should be allowed to input, update and view detailed data for plants, activity performance, resource utilization and environmental impacts and generate an analysis report.

**1.3 Business rules**

**1.3.1 Inventory Management Module**

1. One **Plant** is included in one and only one **Order**. One Order can include one or many **Plant**.
2. One **Plant** will be handled by one and only one **Admin**. One **Admin** handles one or many **Plant**.
3. One **Customer** can purchase one or many **Order**. One **Order** can be purchased by one and only one **Customer**.

##### 1.3.2 Biodiversity Tracking Module

1. One **Plant Record** contains one or many information of **Growth Log**. One **Growth Log** is contained in one or many **Plant Record**.
2. One **Plant Record** contains one or many **Plant Locations**. One **Plant Location** is contained in one or many **Plant Record**.
3. One **Plant** is recorded in one and only one **Plant Record**. One **Plant Record** has recorded one or many **Plant**.

##### 1.3.3 Waste Reduction & Sustainability Module

1. One **Record\_Waste** records one and only one **Activity**.One **Activity** recorded by one or many **Record\_Waste**.
2. One **Record\_Waste** takes in one and only one **RecyclingCenter**. One **RecyclingCenter** takes in one or many **Record\_Waste**.
3. One **Record\_Waste** requires one or many **Sustainability\_Goal**. One **Sustainability\_Goal** required by one or many **Record\_Waste**.

##### 1.3.4 Environmental Impact Module

1. One **Plant** involves one or many **Activity.** One **Activity** involved inone or many **Plant**.
2. One **Activity** causes one or many **Environmental Implication.** One **Environmental Implication** caused by one or many **Activity.**
3. One **Analysis Report** records one or many **Environmental Implication**. One **Environmental Implication** recorded by one or many **Analysis Report.**
4. One **Activity** produces one or many **Record\_Waste**. One **Record\_Waste** produced by one and only **Activity**. `

**1.4 Entity relationship modelling**

**1.4.1 Inventory Management Module**

Diagram is shown in the appendix.

Link of ERD for module 1:

<https://app.diagrams.net/#G1hW1VhWLcVCzmR-c6vLWTmeJlISCyt1f7#%7B%22pageId%22%3A%22_4d9_75VsXDnu6QXlBz0%22%7D>

Explanation: The Inventory Management module consists of 6 entities, which are Plant, Order, Customer, Admin, Activity and Plant\_Record.

Relationship:

* Plant and Order have a one-to-many relationship.(1:M)
* This is a mandatory relationship since every instance of the Plant entity is involved with the Order entity.
* The relationship is a weak relationship because the primary key of the Order entity does not inherit the primary key of the Plant entity.
* Customer and Order have a one-to-many relationship. (1:M)
* This is a mandatory relationship since every instance of the Customer entity is involved with the Order entity.
* The relationship is a weak relationship because the primary key of the Customer entity does not inherit the primary key of the Order entity.
* Admin and Plant have a one-to-many relationship. (1:M)
* This is a mandatory relationship since every instance of the Admin entity is involved with the Plant entity.
* The relationship is a weak relationship because the primary key of the Admin entity does not inherit the primary key of the Plant entity.
* Activity and Plant have a one-to-many relationship. (1:M)
* This is a mandatory relationship since every instance of the Activity entity is involved with the Plant entity.
* The relationship is a weak relationship because the primary key of the Plant entity does not inherit the primary key of the Activity entity.
* Plant\_Record and Plant have a one-to-many relationship. (1:M)
* This is a mandatory relationship since every instance of the Plant\_Record entity is involved with the Plant entity.
* The relationship is a weak relationship because the primary key of the Plant\_Record entity does not inherit the primary key of the Plant entity.

Key Constraints:

1. Plant entity

* contain primary key named Plant\_ID, which is uniquely identified for all the plants
* containing attributes like Plant\_Name, Plant\_Category, Plant\_Stock\_Quantity, Plant\_Price, Plant\_Last\_Updated\_Time and Plant\_Species.
* contain three foreign keys named Order\_ID that referred to the Order entity, Admin\_ID that referred to the Admin entity and Plant\_Record\_ID that referred to the Plant\_Record entity.

1. Order entity

* contain primary key named Order\_ID, which is uniquely identified for all the orders
* containing attributes like Order\_Date, Order\_Total\_Amount and Order\_Status.
* contain one foreign key named Customer\_ID that referred to the Customer entity.

1. Customer entity

* contain primary key named Customer\_ID, which is uniquely identified for all the customers
* containing attributes like Customer\_Name, Customer\_Email, Customer\_Phone, Customer\_Address and Customer\_Date\_Joined.
* no foreign key for this entity

1. Admin entity

* contain primary key named Admin\_ID, which is uniquely identified for all the admins
* containing attributes like Admin\_Name, email, Admin\_Phone, Admin\_Email, Admin\_Address and Admin\_Date\_Joined.
* no foreign key for this entity

1. Plant\_Record entity

* contain a primary key named Plant\_Record\_ID, which is uniquely identified for all the records.
* containing attributes like Planting\_Date.
* no foreign key for this entity (Note that the entity Plant\_Record does not appear in this module scope)

1. Activity entity

* contain a primary key named Activity\_ID, which is uniquely identified for all the activities.
* containing attributes like Activity \_Name and Activity\_Description.
* There is one foreign key named Plant\_ID that referred to the Plant entity. (Note that the entity Activity does not appear in this module scope)

Clarification:

* The Admin\_ID uniquely identifies each admin and links with the Plant entity to manage the inventory management system. Attributes like Admin\_Name, Admin\_Phone, Admin\_Email, Admin\_Address and Admin\_Date\_Joined ensure proper communication between admin and plants.
* The Admin\_ID serves as the unique identifier and connects with the Plant entity. Attributes like Plant\_Name, Plant\_Category and Plant\_Species help in personalizing and managing the plant inventory.
* The Customer\_ID uniquely identifies each admin and links with the Order entity to handle the orders from customers. Attributes like Customer\_Name, Customer\_Phone, Customer\_Email, Customer\_Address and Customer\_Date\_Joined ensure proper communication between customer and orders.
* The Customer\_ID serves as the unique identifier and connects with the Order entity. Attributes like Order\_Date, Order\_Total\_Amount and Order\_Status help customers to track their orders.
* The admin tracks the customer’s order and its details like, including the Unit\_No. and Room\_No. to ensure proper resource allocation and scheduling of appointments within designated spaces.
* The Plant\_ID ensures that each appointment is uniquely identifiable. Foreign keys (Admin\_ID and Order\_ID) define the involved admin, customer, order and plant. This combine and let the inventory management works so that admin can keep track to the plant inventory with the attributes (Plant\_Stock\_Quantity and Plant\_Last\_Updated\_Time)
* The Activity\_ID uniquely identifies each activity and links with the Plant entity to track what activities used to keep plant health. Attributes like Activity\_Name, Activity\_Description, ensure proper relationship between plant and activity so that admin understands.
* The Activity\_ID serves as the unique identifier and connects with the Plant entity. Attributes like Plant\_Name, Plant\_Category, Plant\_Stock\_Quantity, Plant\_Price, Plant\_Last\_Updated\_Time and Plant\_Species to help admin to track the activities involved.
* The Plant\_Record\_ID uniquely identifies each activity and links with the Plant entity to record every plant inside the plant inventory to ease the tracking and growth log job. Attributes like Planting\_Date, ensure proper relationship between plant and plant record so that systems are readable by admin.
* The Plant\_Record\_ID serves as the unique identifier and connects with the Plant entity. Attributes like Planting\_Date to record down information of the plant so that admin can have more data to track and analyze.
* Tracks task-specific information linked to each entity. The Plant\_ID ensures the admin can oversee the whole module and ensure proper execution with efficiency.

**1.4.2 Biodiversity Tracking Module**

Diagram is shown in the appendix.

Link of ERD for module 2:

<https://app.diagrams.net/#G1hW1VhWLcVCzmR-c6vLWTmeJlISCyt1f7#%7B%22pageId%22%3A%22FXRhKRxnGrhVWz5rwQ9f%22%7D>

Explanation: The Biodiversity Tracking Module consists of 6 entities which are Plant\_Record, Plant\_Record\_Location, Plant\_Location, Plant\_Growth\_Record, Growth\_Log and Plant.

Relationship:

* Plant\_Record and Plant\_Location have a many-to-many relationship (M:N). In order to resolve this many-to-many relationship (M:N), an entity called Plant\_Record\_Location has been used as a bridge entity to link Plant\_Record entity and Plant\_Location entity.
* Plant\_Record\_Location has a weak relationship with both Plant\_Record entity and Plant\_Location entity since it inherits both primary keys from Plant\_Record entity and Plant\_Location entity but not as a primary key in Plant\_Record\_Location entity.
* Plant\_Record\_Location is a weak entity as it is existent-dependent on Plant\_Record entity and Plant\_Location entity, indicating that it will only exist when it is associated with the occurrence of Plant\_Record entity and Plant\_Location entity.
* Plant\_Record and Growth\_Log have a many-to-many relationship (M:N). In order to resolve this many-to-many relationship (M:N) an entity called Plant\_Growth\_Record has been used as a bridge entity to link Plant\_Record entity and Growth\_Log entity.
* Plant\_Growth\_Record has a weak relationship with both Plant\_Record entity Growth\_Log entity since it inherits both primary keys from Plant\_Record entity and Growth\_Log entity but not as a primary key in Plant\_Growth\_Record entity.
* Plant\_Growth\_Record is a weak entity as it is existent-dependent on Plant\_Record entity and Growth\_Log entity, indicating that it will only exist when it is associated with the occurrence of Plant\_Record entity and Growth\_Log entity.

Key Constraints:

1. Plant\_Record

* contains a primary key named Plant\_Record\_ID, which uniquely identifies all plant records
* contains attribute Planting\_Date (date the plant was planted)
* no foreign key for this entity

1. Plant\_Location

* contains a primary key named Plant\_Location\_ID, which uniquely identifies all plant locations
* contains attributes Plant\_Location\_Name (name of place where plant is planted), Coordinates (geographical coordinates of the location) and Environment\_Type (environment type such as “tropical” and “desert”)
* no foreign key for this entity

1. Plant\_Record\_Location

* contains a primary key named Plant\_Record\_Location\_ID, which uniquely identifies all plant record locations
* contains two foreign keys, Plant\_Record\_ID, which refers to the Plant\_Record entity, and Plant\_Location\_ID, which refers to the Plant\_Location entity

1. Growth\_Log

* contains a primary key named Growth\_Log\_ID, which uniquely identifies all growth logs
* contains attributes Date (date of growth log entry), Height (height of plant), Health\_Condition (health condition of plant) and Stem\_Circumference (stem circumference of plant).
* no foreign keys for this entity

1. Plant\_Growth\_Record

* contains a primary key named Plant\_Growth\_Record\_ID, which uniquely identifies records of plant growth
* contains attribute Growth\_Status (status of plant growth, e.g. “Healthy” or “Unhealthy”)
* contains two foreign keys, Plant\_Record\_ID which refers to the Plant\_Record entity and Growth\_Log\_ID, which refers to the Growth\_Log entity

Clarification:

* The Plant\_Record\_ID uniquely identifies each plant record and links with the Plant\_Location entity to manage where plants are grown. Attributes like Planting\_Date allow admins to document the planting history of each plant.
* The Plant\_Location\_ID uniquely identifies each location and connects with the Plant\_Record entity to indicate where specific plants are located. Attributes like Plant\_Location\_Name, Coordinates, and Environment\_Type ensure that the nursery can track environmental conditions for optimal plant growth.
* The Plant\_Record\_Location entity serves as a bridge between Plant\_Record and Plant\_Location to resolve the many-to-many relationship. It contains foreign keys (Plant\_Record\_ID and Plant\_Location\_ID) to link records and locations. This ensures efficient tracking of plants across multiple locations.
* The Plant\_Record\_ID uniquely identifies each plant record and links with the Growth\_Log entity to track the growth and health metrics of each plant. Attributes like Planting\_Date help document the plant's lifecycle.
* The Growth\_Log\_ID uniquely identifies each growth log entry and connects with the Plant\_Record entity to monitor specific growth metrics like Height, Health\_Condition, and Stem\_Circumference.
* The Plant\_Growth\_Record entity serves as a bridge between Plant\_Record and Growth\_Log to resolve the many-to-many relationship. It includes foreign keys (Plant\_Record\_ID and Growth\_Log\_ID) and the attribute Growth\_Status, allowing admins to monitor the health and development of plants effectively.
* The Plant\_Record\_ID uniquely identifies each plant record and connects with other entities to maintain comprehensive data about the plants. Attributes like Planting\_Date enable admins to manage planting schedules and maintain historical records for all plants.
* The Plant\_Location\_ID uniquely identifies each location and connects with the Plant\_Record\_Location entity to manage environmental data. Attributes like Plant\_Location\_Name, Coordinates, and Environment\_Type help admins ensure that plants are grown under optimal conditions tailored to their specific needs.
* The Growth\_Record\_ID uniquely identifies each plant growth record and connects with both the Plant\_Record and Growth\_Log entities. Attributes like Growth\_Status help admins prioritize care for plants based on their health and growth trends.
* The Growth\_Log\_ID uniquely identifies each log entry and links with the Plant\_Growth\_Record entity to monitor detailed growth metrics. Attributes like Date, Height, Health\_Condition, and Stem\_Circumference ensure that admins can regularly track plant development and take corrective measures as needed.

**1.4.3 Waste Reduction & Sustainability Module**

Diagram is shown in the appendix.

Link of ERD for module 3:

<https://app.diagrams.net/#G1hW1VhWLcVCzmR-c6vLWTmeJlISCyt1f7#%7B%22pageId%22%3A%22o6moojxcsV83oM0NshiS%22%7D>

Explanation: TheWaste Reduction and Sustainability module consists of 4 entities, which are Waste, Waste\_Goal, SustainabilityGoal and RecyclingCenter.

Relationship:

* RecyclingCenter and Waste have a one-to-many relationship.(1:M)
* This is a mandatory relationship since every instance of the Waste entity is involved with the RecyclingCenter entity.
* The relationship is a weak relationship because the primary key of the RecyclingCenter entity is not in the primary key of the Waste entity.
* Waste and SustainabilityGoal have a many-to-many relationship. (M:N)
* In order to resolve this many-to-many relationship (M:N), an entity called Waste\_Goal has been used as a bridge entity to link Waste entity and SustainabilityGoal entity.
* Waste\_Goal has a weak relationship with both Waste entity and SustainabilityGoal entity since it inherits both primary keys from Waste entity and SustainabilityGoal entity but not as a primary key in Waste\_Goal entity.
* Waste\_Goal is a weak entity as it is existent-dependent on Waste entity and SustainabilityGoal entity, indicating that it will only exist when it is associated with the occurrence of Waste entity and Sustainability entity.
* Waste and Resource have a one-to-many relationship.(1:M)
* This is a mandatory relationship since every instance of the Source entity is involved with the Waste entity.
* The relationship is a weak relationship because the primary key of the Waste entity is not in the primary key of the Resource entity.

Key Constraints:

1. Waste

* contains a primary key named WasteID, which uniquely identifies all waste records
* contains the attribute Quantity(Quantity of the waste), WasteDate(The date when the waste is recorded) and WasteType( The type of waste).
* contains two foreign keys, Resource\_ID which refers to the Resource entity and CenterID , which refers to the RecyclingCenter entity.

1. RecyclingCenter

* contains a primary key named CenterID, which uniquely identifies all records of recycling centers.
* contains the attribute CenterName(The name of the recycling center), CenterLocation(The location of the recycling center) and Capacity( The capacity of the recycling center).
* no foreign key for this entity

1. Waste\_Goal

* contains a primary key named Waste\_GoalID, which uniquely identifies all waste goals.
* contains two foreign keys, WasteID, which refers to the Waste entity, and GoalID which refers to the SustainabilityGoal entity.

1. SustainabilityGoal

* contains a primary key named GoalID, which uniquely identifies all records of sustainability goals.
* contains the attribute GoalDesc(Description of the goal), GoalDeadline(The deadline that is set to achieve the goal) and GoalProgress( The real-time progress of the goal).
* no foreign key for this entity

Clarification:

* The WasteID uniquely identifies each waste record and links with the SustainabilityGoal entity to reduce the waste. Attributes like Quantity, WasteDate and WasteType allow admins to identify the ways to reduce the waste.
* The GoalID uniquely identifies records of sustainability goals and connects with the Waste entity to ensure the waste is reduced. Attributes like GoalDesc, GoalDeadline, and GoalProgress ensure the progress is documented and the waste is successfully reduced before the deadline.
* The Waste\_Goal entity serves as a bridge between Waste and SustainabilityGoal to resolve the many-to-many relationship. It contains foreign keys (WasteID and GoalID) to link the records of waste and the sustainability goals .
* The WasteID uniquely identifies each waste record and links with the RecyclingCenter entity to manage where the waste goes to. Attributes like Quantity, WasteDate and WasteType allow admins to determine which recycling center the waste goes to.
* The CenterID uniquely identifies the record of recycling centers and links with the Waste entity to distribute the waste efficiently. Attributes like CenterName, CenterLocation and Capacity allow admins to select which recycling center the waste goes to .

**1.4.4 Environmental Impact Module**

Diagram is shown in the appendix.

Link of ERD for module 4:

<https://app.diagrams.net/#G1hW1VhWLcVCzmR-c6vLWTmeJlISCyt1f7#%7B%22pageId%22%3A%22SaSM2R1Ws3GtRAjAiAud%22%7D>

Relationship:

* Plant and Activity have a one-to-many relationship (M:N). To resolve this many-to-many relationship, we need to set a bridge entity called Plant\_Activity to link both entities which are Activity and Plant.
* This is a mandatory relationship since every instance of the Plant entity is involved with the Activity entity.
* The relationship is a weak relationship because the Plant\_Activity bridge entity has its own primary key.
* Activity and Environmental\_Implication have many-to-many relationship (M:N). To resolve this many-to-many relationship, we need to set a bridge entity called Activity\_Env\_Imp to link both entities which are Activity and Environmental\_Implication.
* This is a mandatory relationship since every instance of the Activity entity is involved with the Environmental\_Implication entity.
* The relationship is a weak relationship because the Activity\_Env\_Imp bridge entity has its own primary key.
* Analysis\_Report and Environmental\_Implication have a one-to-many relationship (M:N). To resolve this many-to-many relationship, we need to set a bridge entity called Analysis\_Env\_Imp to link both entities which are Analysis\_Report and Environmental\_Implication.
* The relationship is a weak relationship because the both entities do not influence the primary key of either entity.
* Activity and Record\_Wast have a one-to-many relationship (1:M).
* This is a mandatory relationship since every instance of the Activity entity is involved with the Record\_Wast entity.
* The relationship is a weak relationship because the both entities do not influence the primary key of either entity.

Key Constraints:

1. Activity entity
   * contains a primary key named Activity\_ID.
   * contains Plant\_ID as foreign key which referred to the Plant entity.
   * containing attributes like Activity\_Name, and Activity\_Description.
2. Plant\_Activity entity
   * contains a primary key named Plant\_Activity\_ID.
   * contains Plant\_ID as a foreign key which refers to the Plant entity.
   * contains Activity\_ID as a foreign key which refers to the Activity entity.
   * containing attributes like Plant\_Activity\_Date and Plant\_Activity\_Quantity.
3. Environmental\_Implication entity
   * contains a primary key named Env\_Imp\_ID.
   * contains Analysis\_Report\_ID as a foreign key which referred to the Analysis\_Report entity.
   * containing attributes like Env\_Imp\_Name, Env\_Imp\_Unit, Env\_Imp\_Threshold, and Env\_Imp\_Monitor\_Period.
4. Analysis\_Report entity
   * contains a primary key named Analysis\_Report\_ID.
   * containing attributes like Analysis\_Report\_Date, and Analysis\_Report\_Summary.
5. Analysis\_Env\_Imp entity
   * contains a primary key named Analysis\_Env\_Imp\_ID.
   * contains Analysis\_Report\_ID as a foreign key which refers to the Analysis\_Report entity.
   * contains Env\_Imp\_ID as a foreign key which refers to the Environmental\_Implication entity.
6. Activity\_Env\_Imp entity
   * is a bridge entity.
   * contains a primary key named Activity\_Env\_Imp\_ID.
   * contains Env\_Imp\_ID as a foreign key which refers to the Environmental\_Implication entity and Activity\_ID as a foreign key which refers to the Activity entity.
   * contains attributes like Env\_Imp\_Type and Env\_Imp\_Measurement.

Clarification:

* The Plant\_ID uniquely identifies each plant and links with the Activity entity to track which activity is performed for which plant. Attributes like Plant\_Name, Plant\_Category, Plant\_Stock\_Quantity, Plant\_Price, Plant\_Last\_Updated\_Time, Plant\_Species ensure proper connection between Plant entity and Activity entity and indirectly maintain connection for resource management, measure environmental implications and generate analysis reports.
* The Record\_WasteID uniquely identifies each resource waste management and links with the Resource entity to track which activity is performed for which plant. Attributes like Plant\_Name, Plant\_Category, Plant\_Stock\_Quantity, Plant\_Price, Plant\_Last\_Updated\_Time, Plant\_Species ensure proper connection between Plant entity and Activity entity.
* The Analysis\_Report\_ID uniquely identifies each analysis report and links to the Environmental Implication entity to measure and record all the environmental implications caused by carrying out activities to the plants in the nursery. The attributes like Analysis\_Report\_Date and Analysis\_Report\_Summary are useful to generate a summary of the analysis report for each environmental implication.
* The Env\_Imp\_ID uniquely identifies each recorded environment impact. This links to the bridge entity named Activity\_Env\_Imp to track which activities contribute to specific environmental implications. The attributes like Env\_Imp\_Name, Env\_Imp\_Unit, Env\_Imp\_Threshold to specify the type of the impacts and its allowable level and Env\_Imp\_Monitor\_Period is used to track the monitoring duration of the analysis.
* The Activity\_ID links to the bridge entity named Activity\_Env\_Imp to monitor the environmental implications caused by the activity.
* Analysis\_Env\_Imp\_ID is a primary key that uniquely identifies records in the Analysis\_Env\_Imp entity, which associates specific environmental implications with analysis reports. It helps manage the relationships between these two entities effectively.

**1.4.5 Overall ERD**

Diagram is shown in the appendix.

Link of ERD for overall module:

<https://drive.google.com/file/d/1hW1VhWLcVCzmR-c6vLWTmeJlISCyt1f7/view?usp=drive_link>

<https://app.diagrams.net/#G1hW1VhWLcVCzmR-c6vLWTmeJlISCyt1f7>

Explanation: The overall ERD consists of 4 modules including Inventory Management module, Biodiversity Tracking module, Waste Reduction & Sustainability module and Environmental Impact module. There are a total 19 entities for these 4 modules like Plant, Admin, Order, Customer, Plant\_Record, Plant\_Record\_Location, Plant\_Location, Plant\_Growth\_Record, Grwoth\_Log, Waste, Waste\_Goal, SustainabilityGoal, RecyclingCenter, Activity, Activity\_Resource, Resource, Activity\_Env\_Imp, Environmental\_Implication and Analysis\_Report. There are 18 relationships that exist inside the 4 modules.

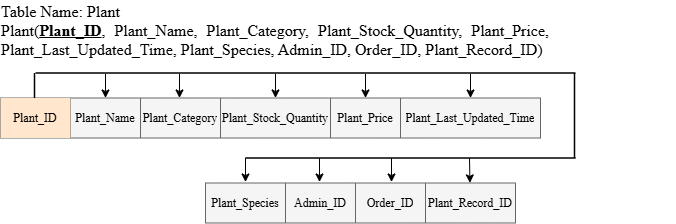
**2.0 Normalization**

Specify the highest normal form all modules should achieve and justify why the normal form is selected. Explain if each table has achieved the selected normal form. If a table is not in the desired normal form, show the normalization steps.

**2.1 Inventory Management Module**

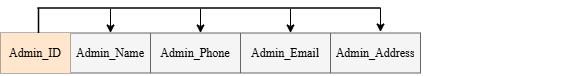
**Table Name: Plant**

Plant **(Plant\_ID,** Plant\_Name, Plant\_Category, Plant\_Stock\_Quantity, Plant\_Price, Plant\_Last\_Updated\_TIme, Plant\_Species, Admin\_ID, Order\_ID, Plant\_Record\_ID

****

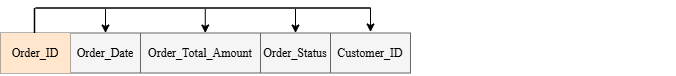
**Table name: Admin**

Admin ( **Admin\_ID**, Admin\_Name, Admin\_Phone, Admin\_Email, Admin\_Address)

****

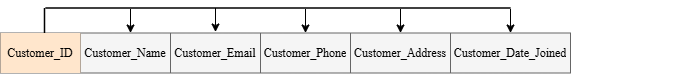
**Table name: Order**

Order (**Order\_ID**, Order\_Date, Order\_Total\_Amount, Order\_Status, Customer\_ID)

****

**Table name: Customer**

Customer (**Customer\_ID**, Customer\_Name, Customer\_Email, Customer\_Phone, Customer\_Address, Customer\_Date\_Joined)

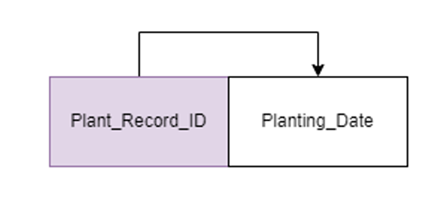
****

If a table is in 3NF and it contains only one candidate key, 3NF and BCNF are equivalent. Since every determinant is a candidate key (composite key is counted as one determinant / candidate key), the table is in 3NF forms, at the same time it is also in BCNF form.

**2.2 Biodiversity Tracking Module**

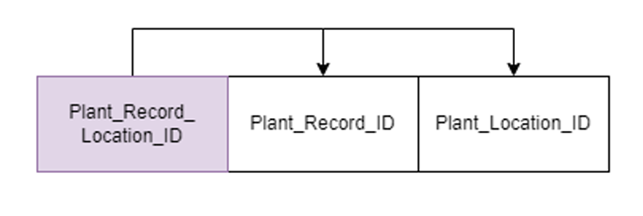
**Table Name: Plant\_Record**

Plant\_Record (**Plant\_Record\_ID**, Planting\_Date)



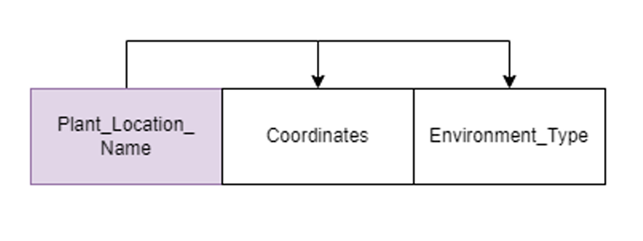
**Table Name: Plant\_Record\_Location**

Plant\_Record\_Location (**Plant\_Record\_Location\_ID**, Plant\_Record\_ID, Plant\_Location\_ID)



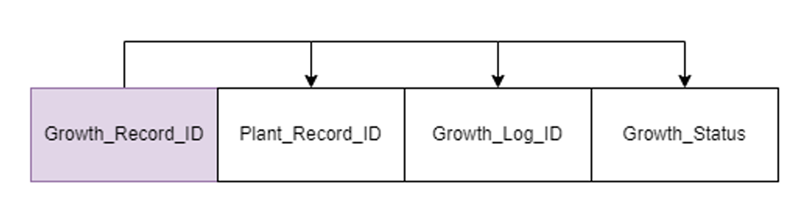
**Table Name: Plant\_Location**

Plant\_Location (**Plant\_Location\_Name**, Coordinates, Environment\_Type)



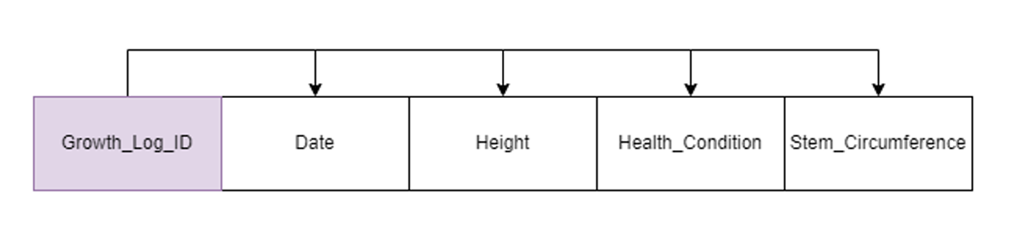
**Table Name: Plant\_Growth\_Record**

Plant\_Growth\_Record (**Growth\_Record\_ID**, Plant\_Record\_ID, Growth\_Log\_ID, Growth\_Status)



**Table Name: Growth\_Log**

Growth\_Log (**Growth\_Log\_ID**, Date, Height, Health\_Condition, Stem\_Circumference)

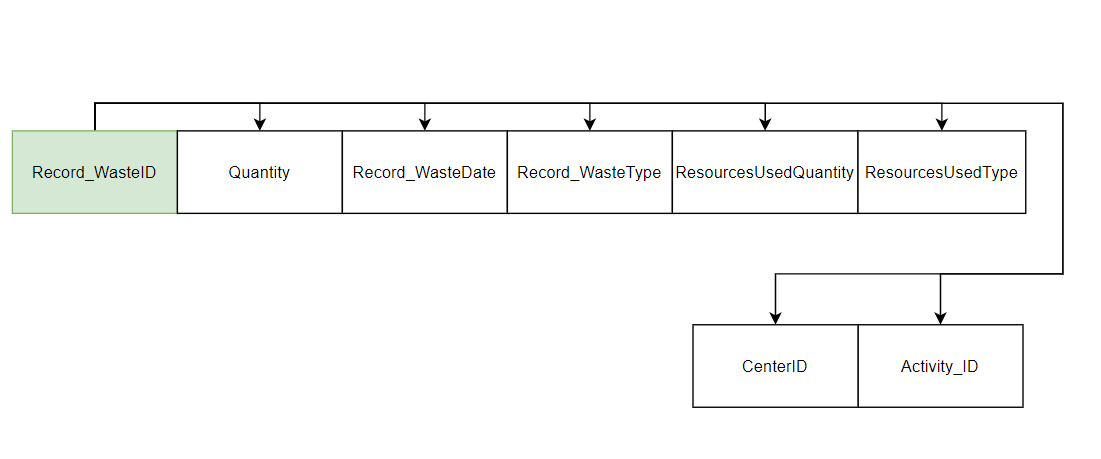


If a table is in 3NF and it contains only one candidate key, 3NF and BCNF are equivalent. Since every determinant is a candidate key (composite key is counted as one determinant / candidate key), the table is in 3NF forms, at the same time it is also in BCNF form.

**2.3 Waste Reduction & Sustainability Module**

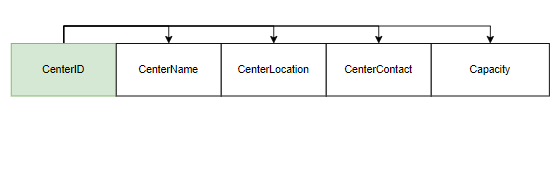
**Table Name: Record\_Waste**

Record\_Waste(**Record\_WasteID**, Quantity,Record\_WasteDate,Record\_WasteType,ResourcesUsedQuantity,ResourcesUsedType,  
Activity\_ID,CenterID)

****

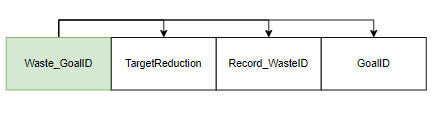
**Table Name: RecyclingCenter**

RecyclingCenter(**CenterID,**CenterName,CenterLocation,CenterContact,Capacity)



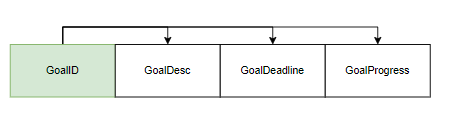
**Table Name: Waste\_Goal**

Waste\_Goal(**Waste\_GoalID,**TargetReduction,Record\_WasteID,GoalID)



**Table Name: Sustainability\_Goal**

Sustainability\_Goal(**GoalID,**GoalDesc,GoalDeadline,GoalProgress)

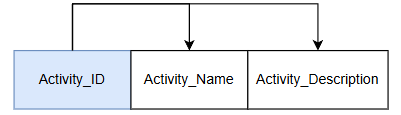


If a table is in 3NF and it contains only one candidate key, 3NF and BCNF are equivalent. Since every determinant is a candidate key (composite key is counted as one determinant / candidate key), the table is in 3NF forms, at the same time it is also in BCNF form.

**2.4 Environmental Impact Module**

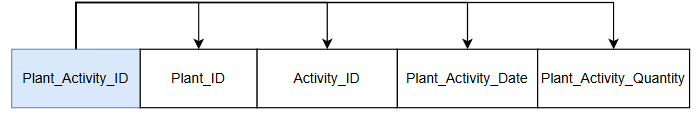
**Table Name: Activity**

Activity (**Activity\_ID**, Activity\_Name, Activity\_Description)



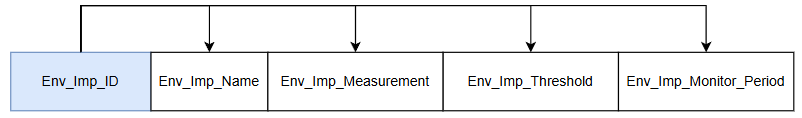
**Table Name: Plant\_Activity**

Plant\_Activity (**Plant\_Activity\_ID**, Plant\_ID, Activity\_ID, Plant\_Activity\_Date, Plant\_Quantity)



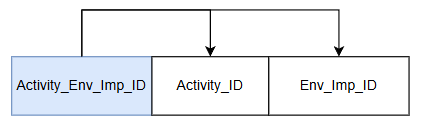
**Table Name: Environmental\_Implication**

Environmental\_Implication (**Env\_Imp\_ID**, Env\_Imp\_Name, Env\_Imp\_Threshold, Env\_Imp\_Monitor\_Period, Env\_Imp\_Measurement)



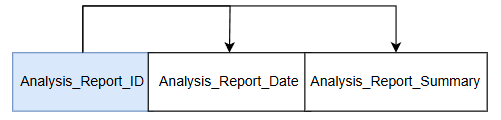
**Table Name: Activity\_Env\_Imp**

Activity\_Env\_Imp (**Activity\_Env\_Imp\_ID**, Activity\_ID, Env\_Imp\_ID)



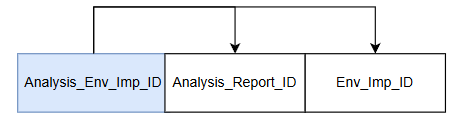
**Table Name: Analysis\_Report**

Analysis\_Report (**Analysis\_Report\_ID**, Analysis\_Report\_Date, Analysis\_Report\_Summary)



**Table Name: Analysis\_Env\_Imp**

Analysis\_Env\_Imp (**Analysis\_Env\_Imp\_ID**, Analysis\_Report\_ID, Env\_Imp\_ID)



If a table is in 3NF and it contains only one candidate key, 3NF and BCNF are equivalent. Since every determinant is a candidate key (composite key is counted as one determinant / candidate key), the table is in 3NF forms, at the same time it is also in BCNF form.

**3.0 Data Dictionary**

**3.1 Inventory Management Module**

Diagram is shown in appendix

Link for data dictionary for module 1: <https://app.diagrams.net/#G12JxtdYg1ihdmj2qA9n3qEh6PwnVKcdk7#%7B%22pageId%22%3A%22tgCWBh-52QgBe2HLFEz0%22%7D>

**3.2 Biodiversity Tracking Module**

Diagram is shown in appendix

**3.3 Waste Reduction & Sustainability Module**

Diagram is shown in appendix

**3.4 Environmental Impact Module**

Diagram is shown in appendix

**4.0 Database implementation**

Internal model representation via SQL commands is expected including the use of triggers and stored procedures for data manipulations.

**4.1 Inventory Management Module**

Table for SQL commands is shown in the appendix.

**4.2 Biodiversity Tracking Module**

Table for SQL commands is shown in the appendix.

**4.3 Waste Reduction & Sustainability Module**

Table for SQL commands is shown in the appendix.

**4.4 Environmental Impact Module**

Table for SQL commands is shown in the appendix.

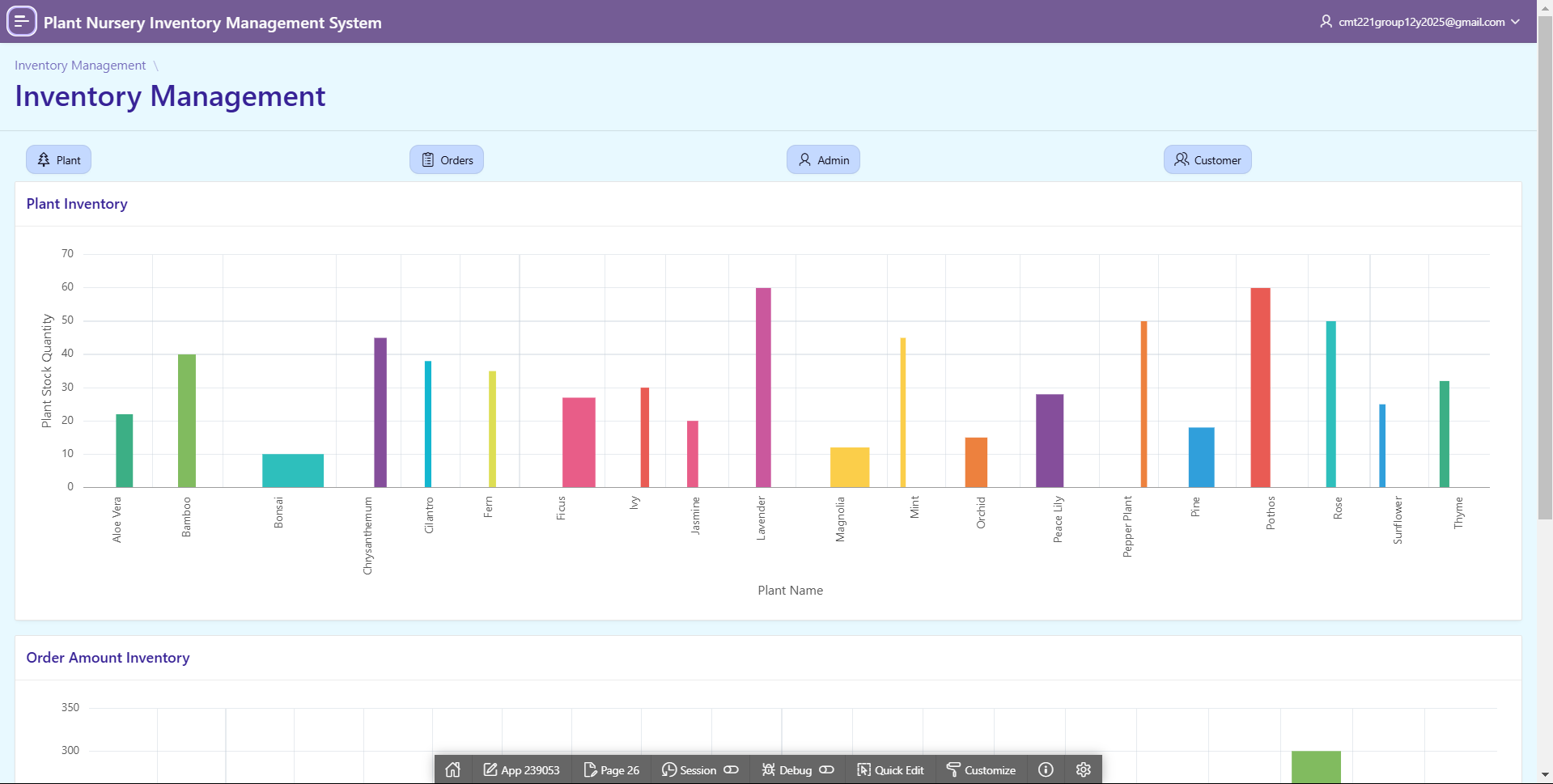
**5.0 Front-end system design and implementation**

a. Graphical user interface design (design standard and notation, screenshots, etc.).

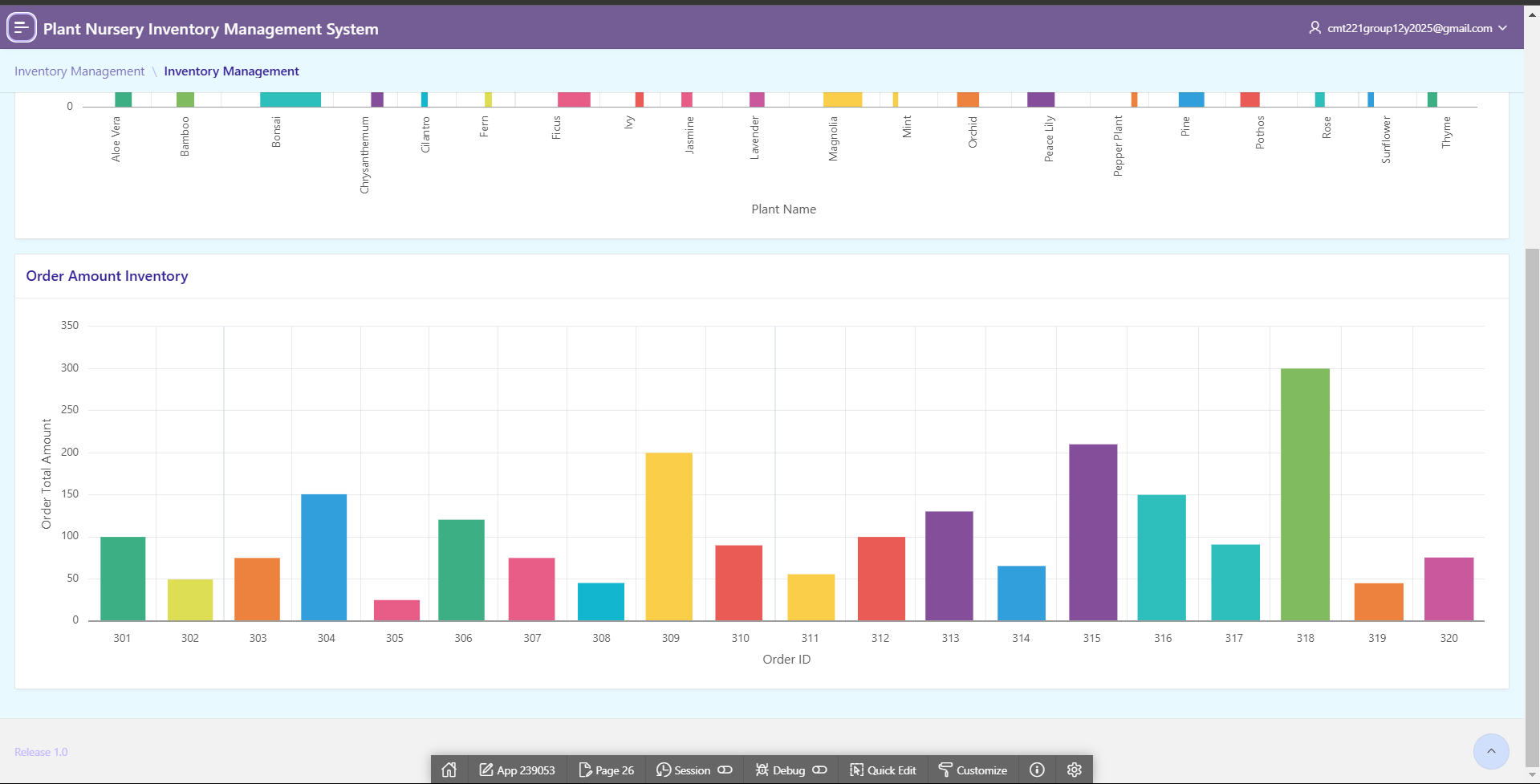
b. A detailed guideline on how you design the front-end system (forms) with proper reasoning is

expected.

c. Report designs.

**5.1 Inventory Management Module**

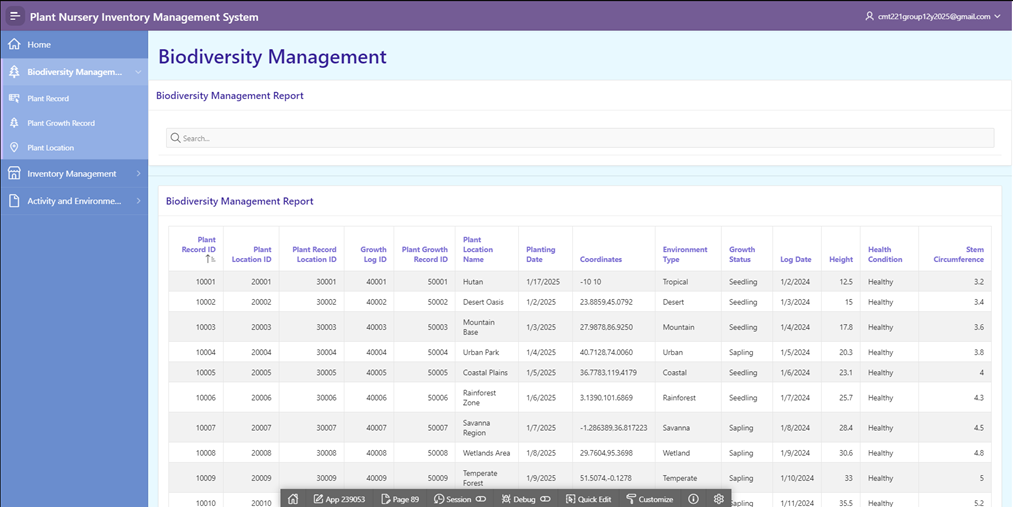
**Figure 5.1 Inventory Management Module Main Page - Part 1**



**Figure 5.2 Inventory Management Module Main Page - Part 2**

This is the Inventory Management Main Page. It shows the chart for plant inventory and order amount inventory. Plant inventory chart is about what plant name and its corresponding plant stock quantity. Order amount inventory is about order\_id which represents every order with its order amount inventory. Admin are able to see how much inventory exists for the plant and orders by just looking at the bar charts. For the details of the module, it will show in the appendix.

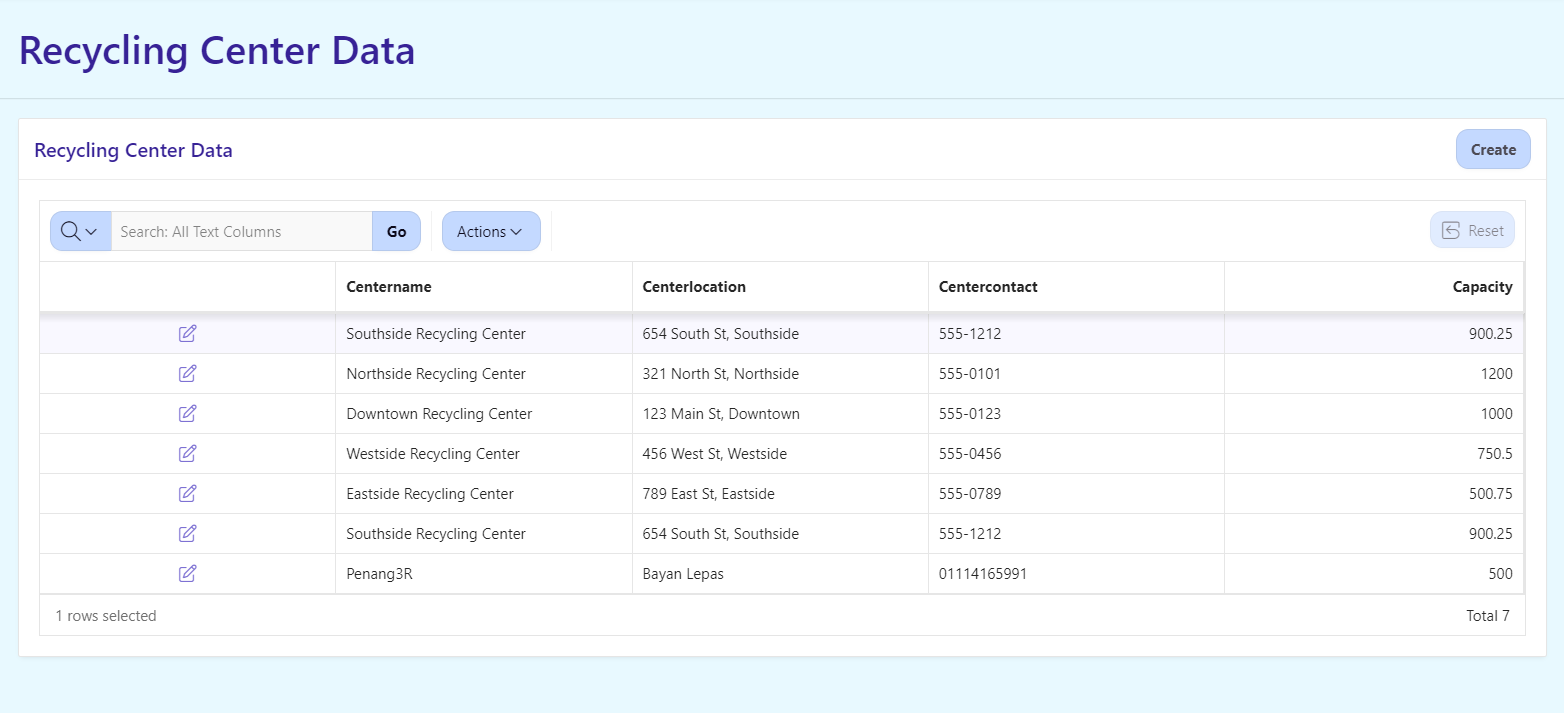
**5.2 Biodiversity Tracking Module**



**Figure 5.3 Biodiversity Management Module Main Page**

This is the Biodiversity Management Module Report Page. It is not editable and values cannot be added to it. It is a summary report of all the information needed for biodiversity management. Thus, the admin will be able to access this page to review any information regarding biodiversity management for quick use. There is also a search function available if the admin would like to search specific information. More information regarding the subpages and functions which consists of Plant Record Report Page, Plant Growth Record Report Page and Plant Location Report Page can be found in the appendix.

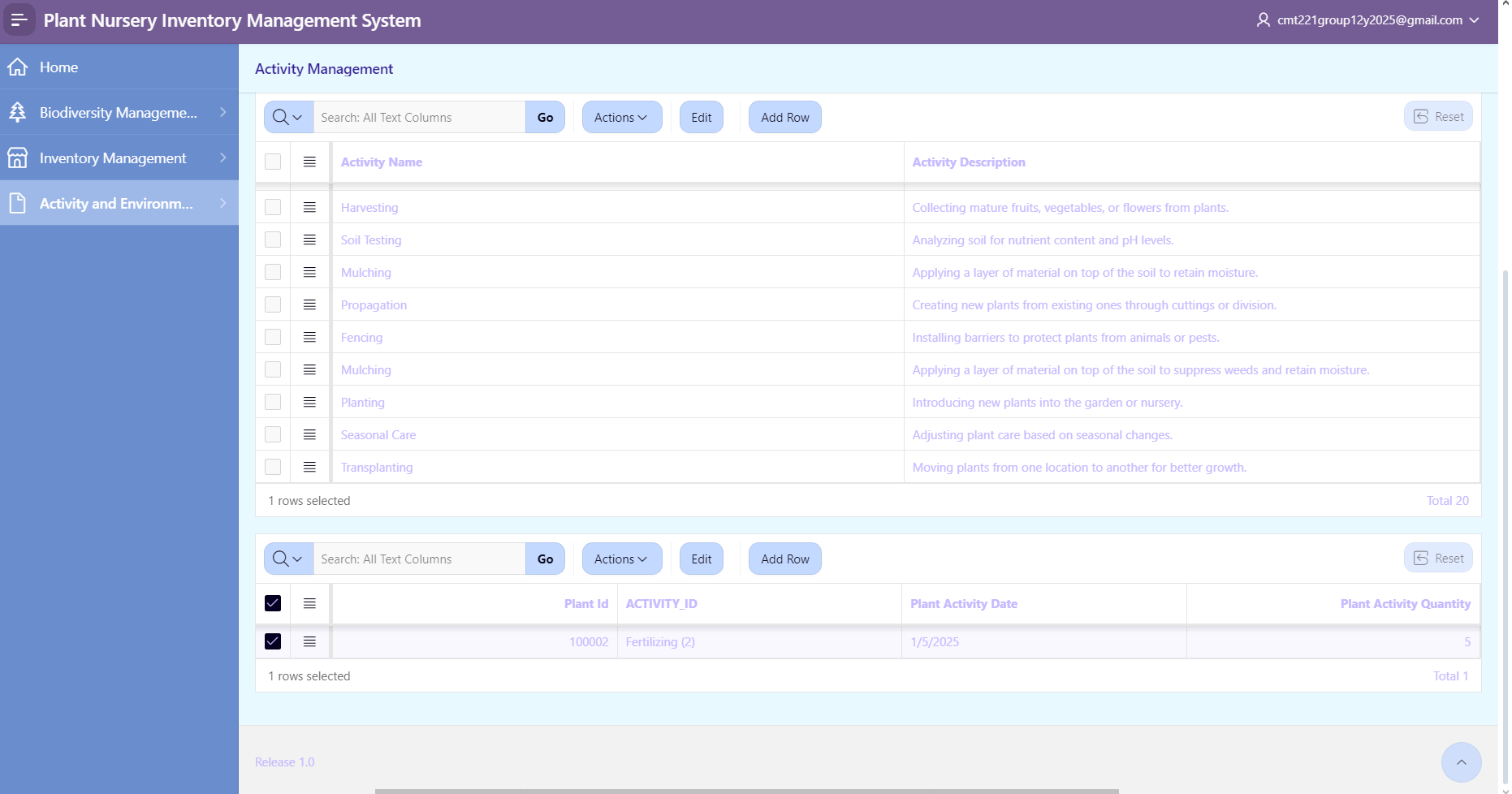
**5.3 Waste Reduction & Sustainability Module**



**Figure 5.4 Waste Reduction & Sustainability Module Main Page**

This is the Waste Management Module Report Page. It is editable and values can be added to it. It is a summary report of all the information needed for Waste management. Thus, the admin will be able to access this page to review any information regarding Waste management for quick use. There is also a search function available if the admin would like to search specific information.

**5.4 Environmental Impact Module**



**Figure 5.5 Activity and Environmental Implication Management Module Page**

This page is the Activity Management page where the administrator can view data of activities performed on the plants in the nursery. This table is useful for the administrator to view which plant is involved in which activities. Administrators can click on any one activity to check what are the plants performed by that activity.

**6.0 Project Problems and Pitfalls**

While there will most certainly be several benefits to YnB’s adoption of the proposed integrated database system, there are also drawbacks that need consideration. Moving from a manual system to an integrated solution is not easy, especially when you have to do it and transfer data from spreadsheets and printed documents. Many steps in the process are difficult and need to be well thought out because any slip may affect the integrity of the system. The same challenges exist if there are multiple modules involved, such as waste management, biodiversity assessment, and inventory control. The integration between such parts must work flawlessly and data interchange must be dependable.

Training and adoption of users is yet another challenge. Manual work may force workforces not familiar with technology to oppose the new system. Staff members may utilize the system inefficiently and therefore, the effectiveness of the system would be very low if proper training and support aren’t offered. Additionally, YnB is likely to face other technology challenges as they expand like limitations relating to growth, performance issues, and system crashes. There is a constant need for knowledge and resources to update changes on data in all modules simultaneously and to ensure everything is fully operational.

Another challenge is the allocation of resources and costs. Development and deployment costs of the system are huge such as user training, hardware, and software. There might create an issue concerning a budget deficit for the organization due to cost overruns or project delays. Moreover, digitalization poses risks to the security and confidentiality of data. The system will contain sensitive information about clients and the surroundings which makes it a target for hackers. Stringent measures such as encryption and access control will be essential to protect data and maintain the confidence of stakeholders.

A further challenge that can arise could be opposition to change as the adoption of a new paradigm may require new practices to be put in place. Employees may not want to go through the hassle of new processes if they feel the new system is difficult to understand. To alleviate this resistance, it is vital to use a proper systematic approach of change communication and properly emphasize the merits of the system. There is also some added complexity in ensuring compliance and reporting activity for sustainability and biodiversity measures. The system must be designed in such a way that the information is captured and reported in accordance with the guidelines as well as YnB's sustainability goals and legal framework.

Through meticulous planning, comprehensive training of users, effective system architecture, and efficient alteration management, YnB can prevent such challenges from surfacing, and put in place the database system which meets its operational and environmental needs.

**7.0 Conclusions/Recommendations/Future Work**

On behalf of YnB, we have learned a lot about database design concepts and how to use Oracle SQL Developer and Apex to apply them due to this project. We’ve addressed important issues such as ineffective inventory keeping and a lack of biodiversity monitoring measures by creating the Plant Nursery Inventory Management System. The significance of integrated database systems in enhancing operational effectiveness and cutting waste has been noticed. We have faced many problems which include our initial lack of experience with Oracle Apex tools and time constraints brought on by conflicting academic obligations; however we fought and persevered and worked together to overcome them. This project's successful completion has improved our ability to solve problems and improved our understanding of how database systems can help with everyday things.

In order to fulfil our environmental objectives and accomplish our SDG, up-to-date data insights are vital. We also advise the staff members to receive training on Oracle SQL Developer and Apex to ensure the system is used efficiently and effectively. Our suggestion is also to incorporate AI-powered tools and predictive analytics to the system. YnB's objectives of ecological protection and operational excellence will be further advanced by these advancements which would offer more accurate forecasting and biodiversity monitoring.

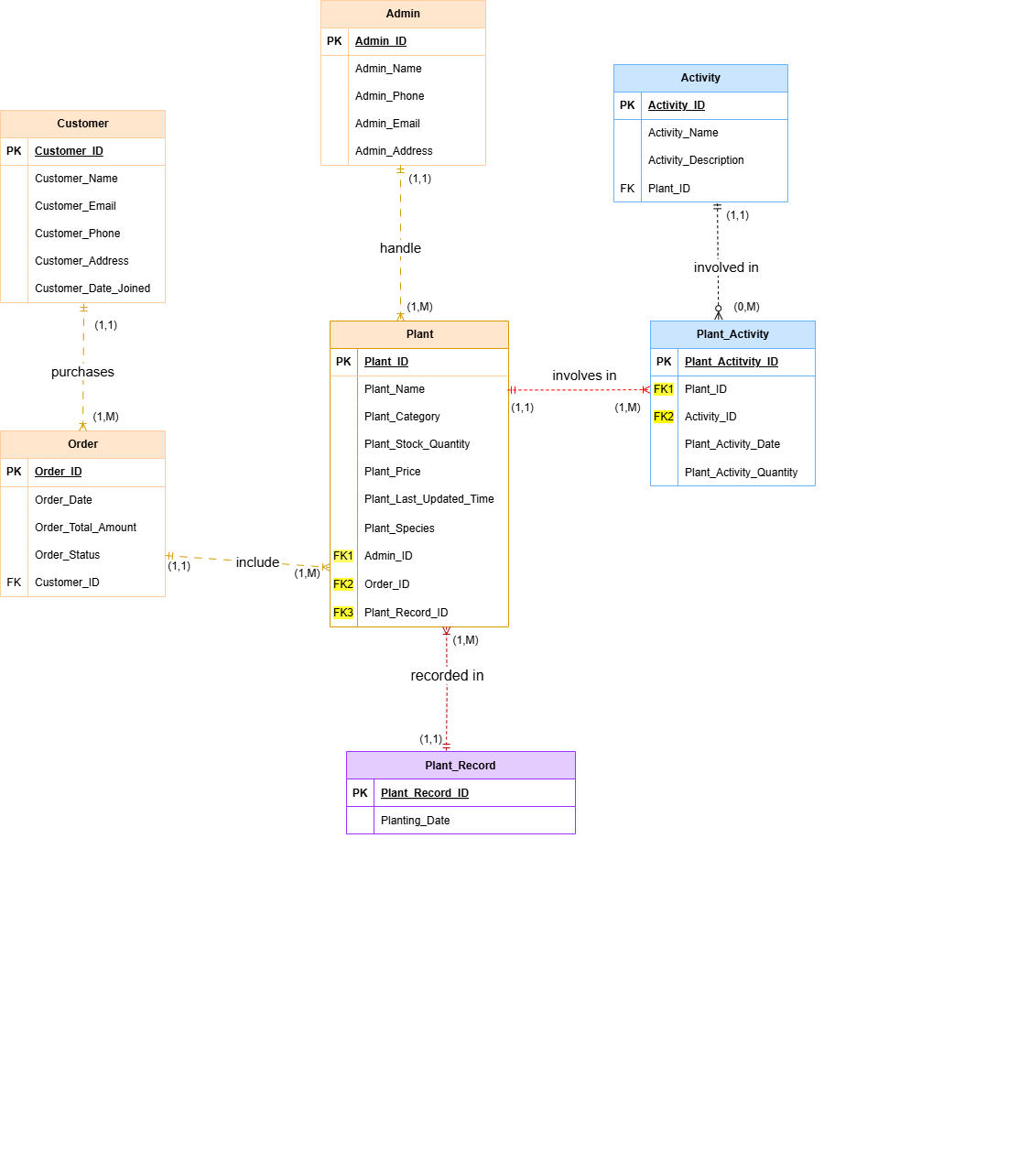
Eventhough many important problems are addressed by the existing system, there is still room for improvement. Usability would definitely be increased by improving the user interface (UI) to make it easier to use for people with lower levels of technical skill. Plant care and resource management may be improved by integrating IoT-enabled sensors to monitor environmental data in real time such as soil temperature and moisture content. Furthermore, creating a mobile application would definitely increase operational flexibility by providing on-the-go access for order management and inventory data. The usability of the system would be enhanced by the addition of professional visualisation capabilities that would allow stakeholders to see our progress toward environmental and waste reduction targets in a straightforward manner.

All in all, this initiative have been a major stride in YnB's goal of being a pioneer in the environmentally friendly operations. Future developments in database design and implementation will be made upon the knowledge and abilities gained during this process as well in due time. In addition to addressing YnB's present issues, the Plant Nursery Inventory Management System has laid the foundation for the framework for accomplishing its long-term sustainability goals.

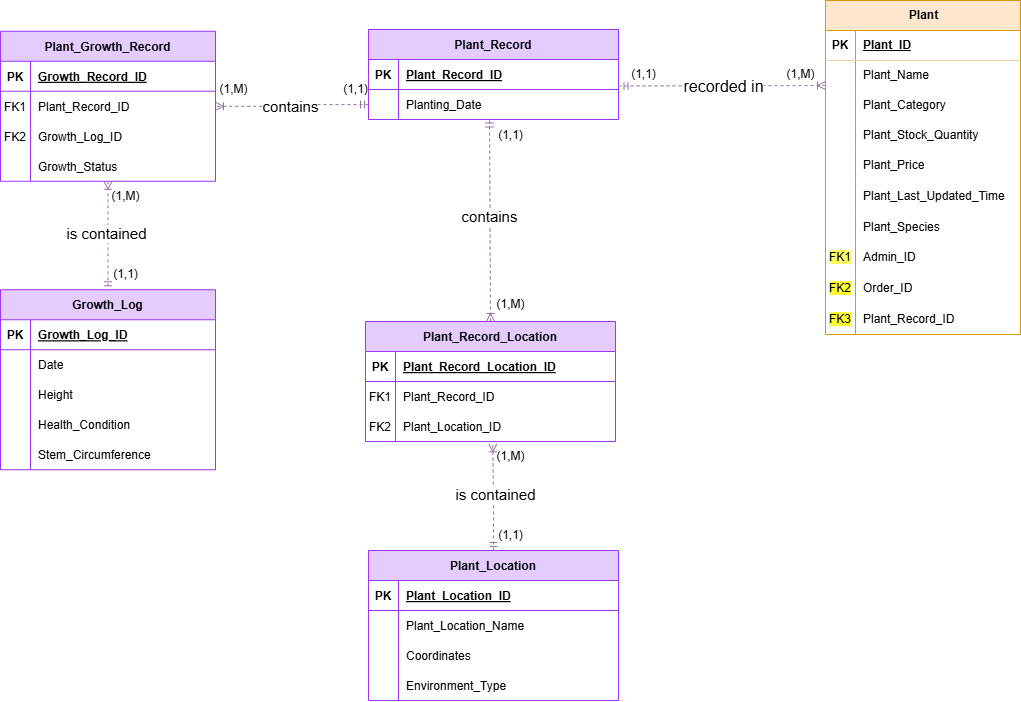
**8.0 References**

1. Connolly, T. M., & Begg, C. E. (1998). *Database Systems: A Practical approach to design, implementation and management*. <https://www.researchgate.net/profile/Thomas_Connolly2/publication/228831514_Database_Systems_A_Practical_Approach_to_Design_Implementation_and_Management/links/0912f510c17cc8db3c000000.pdf>
2. Korth, H. F., & Silberschatz, A. (1980b). *Database System Concepts*. <http://library.upnvj.ac.id/index.php?p=show_detail&id=4402>
3. *DBMS - ER Diagram Representation*. (n.d.). <https://www.tutorialspoint.com/dbms/er_diagram_representation.htm>
4. GeeksforGeeks. (2024, November 29). *Introduction of ER model*. GeeksforGeeks. <https://www.geeksforgeeks.org/introduction-of-er-model/>
5. Bagui, S. S., & Earp, R. W. (2022). *Database Design Using Entity-Relationship Diagrams*. <https://doi.org/10.1201/9781003314455>
6. *MPRise AgriWare | Plant Nursery Management Software*. (n.d.). Mprise Agriware. <https://www.mprise-agriware.com/plant-nursery-software>
7. Acctivate Inventory Management Software. (2024, March 28). *Nursery Inventory Management Software*. QuickBooks Inventory Management Software | Acctivate - QuickBooks Inventory Management Software. <https://acctivate.com/industries/nursery-inventory-management-software/>
8. Nicese, F. P., Azzini, L., Lucchetti, S., Macci, C., Vannucchi, F., Masciandaro, G., Pantani, O. L., Arfaioli, P., Pathan, S. I., Pietramellara, G., & Manzini, J. (2024). Co-Composting of Green Waste and Dredged Sediments Can Reduce the Environmental Impact of the Potted Nursery without Affecting Plant Growth. *Applied Sciences*, *14*(4), 1538. <https://doi.org/10.3390/app14041538>
9. Messelink, G. J., Lambion, J., Janssen, A., & Van Rijn, P. C. J. (2021). Biodiversity in and around Greenhouses: Benefits and Potential Risks for Pest Management. *Insects*, *12*(10), 933. <https://doi.org/10.3390/insects12100933>
10. Hunt, D. (n.d.). *Waste management and disposal in the nursery industry*. <https://nurseryproductionfms.com.au/wp-content/uploads/download-manager-files/Waste-in-the-nursery-industry-Nursery-Paper-Ver3-final-2.pdf>

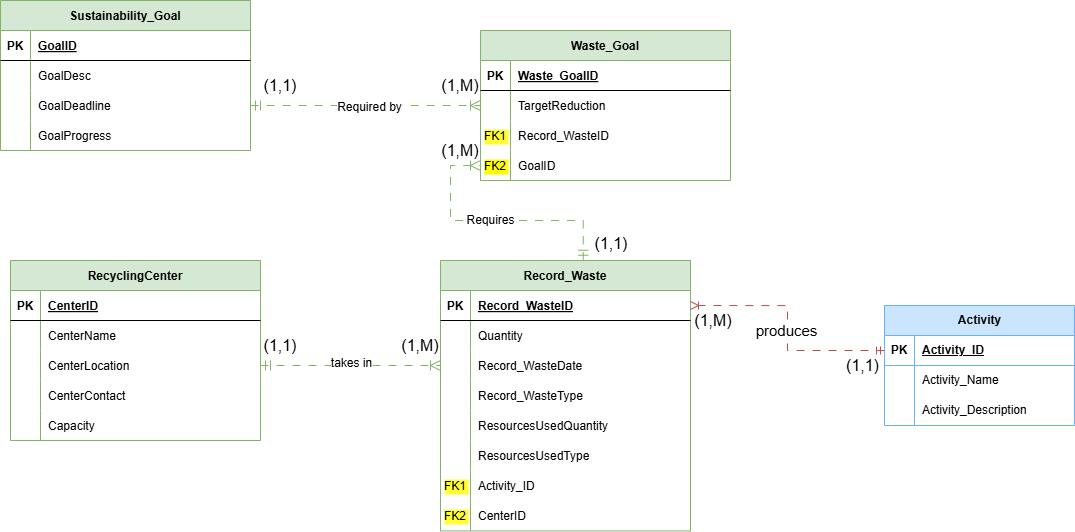
**9.0 Appendix**



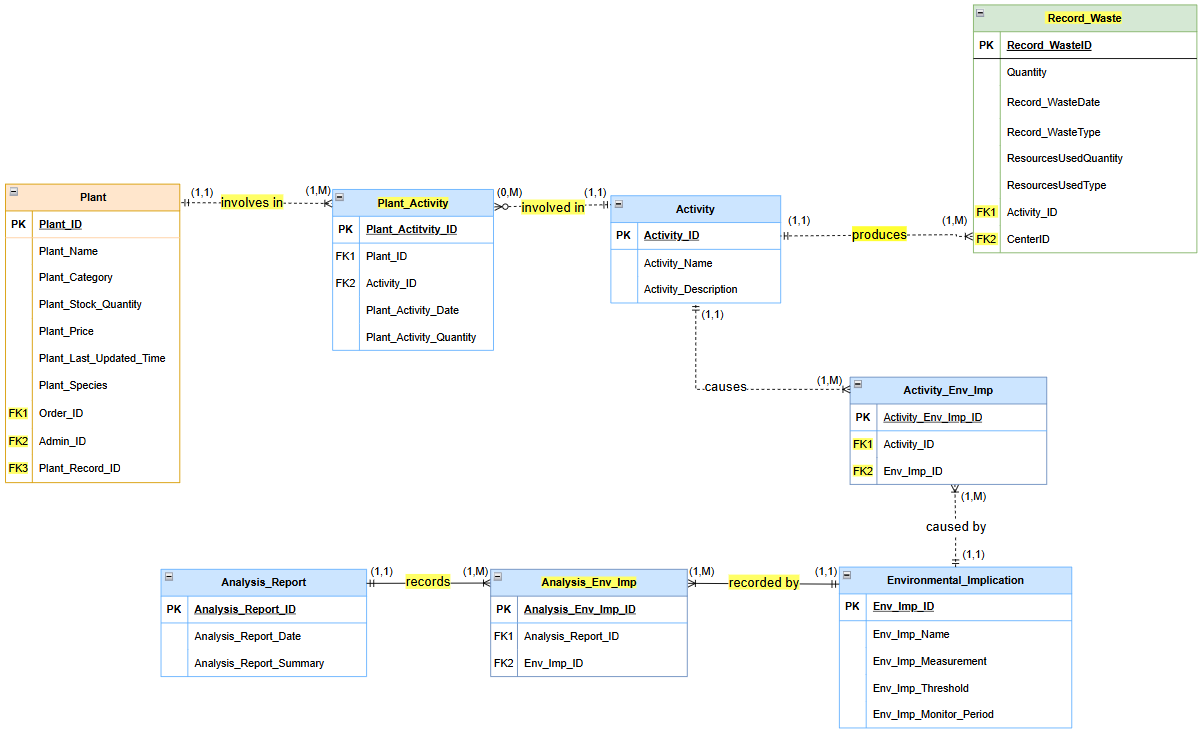
**Figure 1.1: ERD of Inventory Management Module**

****

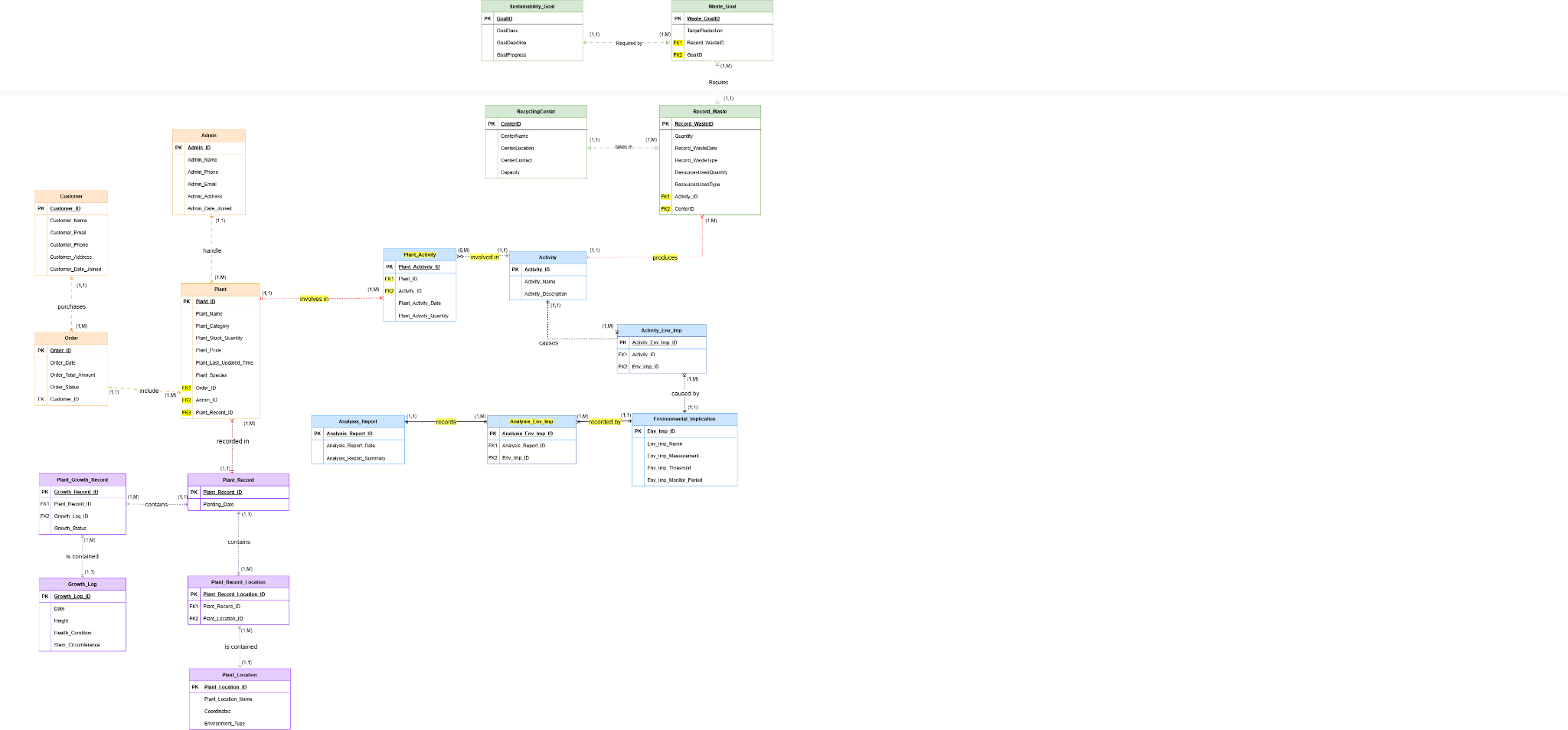
**Figure 1.2: ERD of Biodiversity Tracking Module**

****

**Figure 1.3: ERD of Waste Reduction & Sustainability Module**



**Figure 1.4: ERD of Environmental Impact Module**

****

**Figure 1.5: Overall ERD of all module**



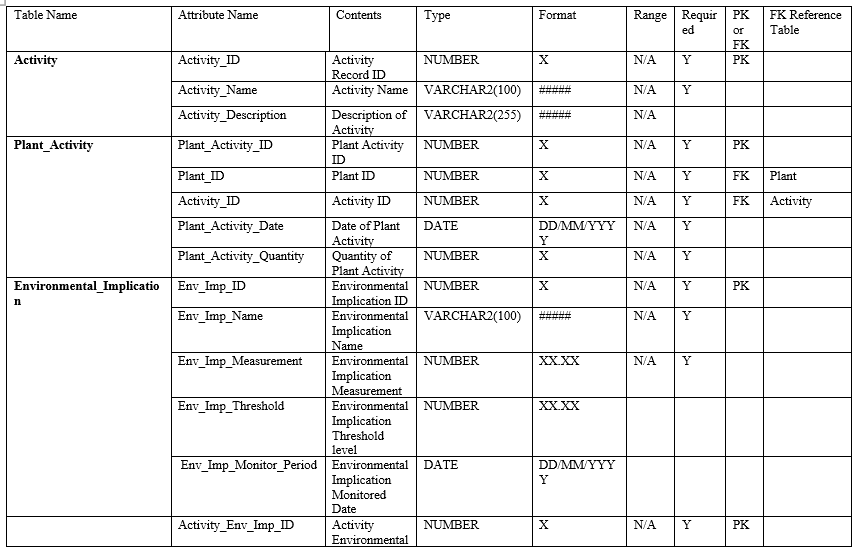
**Figure 3.1 Data dictionary for module 1**

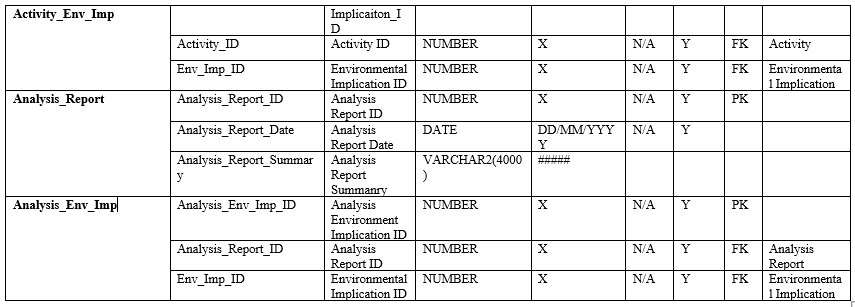


**Figure 3.2 Data dictionary for module 2**

****

**Figure 3.3 Data dictionary for module 3**

****

****

**Figure 3.4 Data dictionary for module 4**

| **Table** | **SQL Statement** |
| --- | --- |
| **Plant** | CREATE TABLE Plant (  Plant\_ID INTEGER PRIMARY KEY,  Plant\_Name VARCHAR2(255) NOT NULL,  Plant\_Category VARCHAR2(100),  Plant\_Stock\_Quantity INTEGER NOT NULL,  Plant\_Price DECIMAL(10, 2) NOT NULL,  Plant\_Last\_Updated\_Time TIMESTAMP NOT NULL,  Plant\_Species VARCHAR2(100),  Admin\_ID INTEGER NOT NULL,  Order\_ID INTEGER NOT NULL,  Plant\_Record\_ID INTEGER NOT NULL,  CONSTRAINT ADMIN\_FK FOREIGN KEY (Admin\_ID)  REFERENCES Admin(Admin\_ID) ON DELETE CASCADE,  CONSTRAINT ORDER\_FK FOREIGN KEY (Order\_ID)  REFERENCES Orders(Order\_ID) ON DELETE CASCADE,  CONSTRAINT PLANT\_RECORD\_FK FOREIGN KEY  (Plant\_Record\_ID)  REFERENCES Plant\_Record(Plant\_Record\_ID) ON DELETE CASCADE  ); |
| **Admin** | CREATE TABLE Admin (  Admin\_ID INTEGER PRIMARY KEY,  Admin\_Name VARCHAR2(255) NOT NULL,  Admin\_Phone VARCHAR2(20) NOT NULL,  Admin\_Email VARCHAR2(255) UNIQUE NOT NULL,  Admin\_Address VARCHAR2(255)  ); |
| **Order** | CREATE TABLE Orders (  Order\_ID INTEGER PRIMARY KEY,  Order\_Date DATE NOT NULL,  Order\_Total\_Amount DECIMAL(10, 2) NOT NULL,  Order\_Status VARCHAR2(50) NOT NULL,  Customer\_ID INTEGER NOT NULL,  CONSTRAINT CUSTOMER\_FK FOREIGN KEY (Customer\_ID)  REFERENCES Customer(Customer\_ID)  ON DELETE CASCADE  ); |
| **Customer** | CREATE TABLE Customer (  Customer\_ID INT PRIMARY KEY,  Customer\_Name VARCHAR2(255) NOT NULL,  Customer\_Email VARCHAR2(255) UNIQUE NOT NULL,  Customer\_Phone VARCHAR2(20) NOT NULL,  Customer\_Address VARCHAR2(255),  Customer\_Date\_Joined DATE NOT NULL  ); |

**Table 4.1 Inventory Management Module**

| Automatically Update Plant\_Last\_Updated\_Time | CREATE OR REPLACE TRIGGER UpdatePlantLastUpdated  BEFORE UPDATE ON Plant  FOR EACH ROW  BEGIN  :NEW.Plant\_Last\_Updated\_Time := SYSDATE;  END;  / |
| --- | --- |
| Prevent Negative Stock in Plant | CREATE OR REPLACE TRIGGER PreventNegativeStock  BEFORE UPDATE ON Plant  FOR EACH ROW  BEGIN  IF :NEW.Plant\_Stock\_Quantity < 0 THEN  RAISE\_APPLICATION\_ERROR(-20001, 'Stock quantity cannot be negative');  END IF;  END;  / |
| Add a New Customer | CREATE OR REPLACE PROCEDURE AddCustomer (  p\_name IN VARCHAR2,  p\_email IN VARCHAR2,  p\_phone IN VARCHAR2,  p\_address IN VARCHAR2,  p\_date\_joined IN DATE  )  AS  BEGIN  INSERT INTO Customer (  Customer\_Name,  Customer\_Email,  Customer\_Phone,  Customer\_Address,  Customer\_Date\_Joined  ) VALUES (  p\_name,  p\_email,  p\_phone,  p\_address,  p\_date\_joined  );  EXCEPTION  WHEN DUP\_VAL\_ON\_INDEX THEN  RAISE\_APPLICATION\_ERROR(-20001, 'Duplicate email address not allowed.');  WHEN OTHERS THEN  RAISE\_APPLICATION\_ERROR(-20002, 'An unexpected error occurred: ' || SQLERRM);  END; |
| Retrieve Plant Details by Category | CREATE OR REPLACE PROCEDURE GetPlantsByCategory (  p\_category IN VARCHAR2,  result\_cursor OUT SYS\_REFCURSOR -- Declare an OUT parameter for the cursor  )  IS  BEGIN  OPEN result\_cursor FOR  SELECT \*  FROM Plant  WHERE Plant\_Category = p\_category;  END; |
| Get customer order | CREATE OR REPLACE PROCEDURE GetCustomerOrders(  p\_CustomerID IN NUMBER,  p\_PageSize IN NUMBER DEFAULT 10,  p\_PageNumber IN NUMBER DEFAULT 1  )  IS  BEGIN  FOR rec IN (  SELECT Order\_ID, Order\_Date, Order\_Total\_Amount, Order\_Status  FROM Orders  WHERE Customer\_ID = p\_CustomerID  OFFSET (p\_PageNumber - 1) \* p\_PageSize ROWS FETCH NEXT p\_PageSize ROWS ONLY  ) LOOP  DBMS\_OUTPUT.PUT\_LINE('Order ID: ' || rec.Order\_ID ||  ', Date: ' || TO\_CHAR(rec.Order\_Date, 'YYYY-MM-DD') ||  ', Total Amount: ' || rec.Order\_Total\_Amount ||  ', Status: ' || rec.Order\_Status);  END LOOP;  END;  / |

**Table 4.2 Inventory Management Module - Trigger and stored procedure**

| **Table** | **SQL Statement** |
| --- | --- |
| **Plant\_Record** | CREATE TABLE Plant\_Record ( Plant\_Record\_ID INT PRIMARY KEY,  Planting\_Date DATE  ); |
| **Plant\_Record\_Location** | CREATE TABLE Plant\_Record\_Location (  Plant\_Record\_Location\_ID INT PRIMARY KEY,  Plant\_Record\_ID INT, Plant\_Location\_ID INT,  FOREIGN KEY (Plant\_Record\_ID) REFERENCES Plant\_Record(Plant\_Record\_ID),  FOREIGN KEY (Plant\_Location\_ID) REFERENCES Plant\_Location(Plant\_Location\_ID)  ); |
| **Plant\_Location** | CREATE TABLE Plant\_Location (  Plant\_Location\_ID INT PRIMARY KEY,  Plant\_Location\_Name VARCHAR(100),  Coordinates VARCHAR(100),  Environment\_Type VARCHAR(50)  ); |
| **Plant\_Growth\_Record** | CREATE TABLE Plant\_Growth\_Record (  Plant\_Growth\_Record\_ID INT PRIMARY KEY,  Plant\_Record\_ID INT,  Growth\_Log\_ID INT,  Growth\_Status VARCHAR(50),  FOREIGN KEY (Plant\_Record\_ID) REFERENCES Plant\_Record(Plant\_Record\_ID),  FOREIGN KEY (Growth\_Log\_ID) REFERENCES Growth\_Log(Growth\_Log\_ID)  ); |
| **Growth\_Log** | CREATE TABLE Growth\_Log (  Growth\_Log\_ID INT PRIMARY KEY,  Log\_Date DATE,  Height DECIMAL(10, 2),  Health\_Condition VARCHAR(50),  Stem\_Circumference DECIMAL(10, 2)  ); |

**Table 4.3 Biodiversity Tracking Module**

| **Function** | **SQL Statement** |
| --- | --- |
| **Automatically Update Plant Location ID** | create or replace TRIGGER trg\_plant\_location\_id  BEFORE INSERT ON Plant\_Location  FOR EACH ROW  BEGIN  :NEW.Plant\_Location\_ID := seq\_plant\_location\_id.NEXTVAL;  END; |
| **Automatically Update Plant Record Location ID** | create or replace TRIGGER trg\_plant\_record\_location\_id  BEFORE INSERT ON Plant\_Record\_Location  FOR EACH ROW  BEGIN  :NEW.Plant\_Record\_Location\_ID := seq\_plant\_record\_location\_id.NEXTVAL;  :NEW.Plant\_Record\_ID := seq\_plant\_record\_id.NEXTVAL;  END; |
| **Automatically Update Growth Log ID** | create or replace TRIGGER trg\_growth\_log\_id  BEFORE INSERT ON Growth\_Log  FOR EACH ROW  BEGIN  :NEW.Growth\_Log\_ID := seq\_growth\_log\_id.NEXTVAL;  END; |
| **Automatically Update Plant Growth Record ID** | create or replace TRIGGER trg\_plant\_growth\_record\_id  BEFORE INSERT ON Plant\_Growth\_Record  FOR EACH ROW  BEGIN  :NEW.Plant\_Growth\_Record\_ID := seq\_plant\_growth\_record\_id.NEXTVAL;  :NEW.Plant\_Record\_ID := seq\_plant\_record\_id.NEXTVAL;  END; |
| **Create Plant Location ID** | CREATE SEQUENCE seq\_plant\_location\_id  START WITH 20001  INCREMENT BY 1  NOCACHE; |
| **Create Plant Record Location ID** | CREATE SEQUENCE seq\_plant\_record\_location\_id  START WITH 30001  INCREMENT BY 1  NOCACHE; |
| **Create Plant Record ID** | CREATE SEQUENCE seq\_plant\_record\_id  START WITH 10001  INCREMENT BY 1  NOCACHE; |
| **Create Growth Log ID** | CREATE SEQUENCE seq\_growth\_log\_id  START WITH 40001  INCREMENT BY 1  NOCACHE; |
| **Create Plant Growth Record ID** | CREATE SEQUENCE seq\_plant\_growth\_record\_id  START WITH 50001  INCREMENT BY 1  NOCACHE; |

**Table 4.4 Biodiversity Tracking Module - Trigger and stored procedure**

| **TABLE** | **SQL STATEMENT** |
| --- | --- |
| **Record\_Waste** | CREATE TABLE Record\_Waste (  CREATE TABLE Record\_Waste (  Record\_WasteID INT PRIMARY KEY,  Quantity DECIMAL(10, 2) NOT NULL,  WasteDate DATE NOT NULL,  ResourceUsedQuantity DECIMAL(10, 2) NOT NULL,  ResourcesUsedType VARCHAR(100) NOT NULL,  WasteType VARCHAR(100) NOT NULL,  ActivityID INT NOT NULL,  CenterID INT NOT NULL,  CONSTRAINT FK\_Activity FOREIGN KEY (ActivityID)  REFERENCES Activity(ActivityID) ON DELETE CASCADE,  CONSTRAINT FK\_Center FOREIGN KEY (CenterID)  REFERENCES RecyclingCenter(CenterID) ON DELETE CASCADE  ); |
| **RecyclingCenter** | CREATE TABLE RecyclingCenter (  CenterID INT PRIMARY KEY,  CenterName VARCHAR(255) NOT NULL,  CenterLocation VARCHAR(255) NOT NULL,  CenterContact VARCHAR(20),  Capacity DECIMAL(10, 2) NOT NULL  ); |
| **Waste\_Goal** | CREATE TABLE Waste\_Goal (  Waste\_GoalID INT PRIMARY KEY,  TargetReduction DECIMAL(10, 2) NOT NULL,  Record\_WasteID INT NOT NULL,  GoalID INT NOT NULL,  CONSTRAINT FK\_RecordWaste FOREIGN KEY (Record\_WasteID)  REFERENCES Record\_Waste(Record\_WasteID) ON DELETE CASCADE,  CONSTRAINT FK\_Goal FOREIGN KEY (GoalID)  REFERENCES SustainabilityGoal(GoalID) ON DELETE CASCADE  ); |
| **SustainabilityGoal** | CREATE TABLE SustainabilityGoal (  GoalID INT PRIMARY KEY,  GoalDesc VARCHAR(255) NOT NULL,  GoalDeadline DATE NOT NULL,  GoalProgress DECIMAL(5, 2) NOT NULL  ); |

**Table 4.5 Waste Reduction & Sustainability Module**

| **TABLE** | **SQL STATEMENT** |
| --- | --- |
| **Automatically Update GoalProgress** | CREATE TRIGGER UpdateGoalProgress  AFTER INSERT ON Waste\_Goal  FOR EACH ROW  BEGIN  UPDATE SustainabilityGoal  SET GoalProgress = GoalProgress + NEW.TargetReduction  WHERE GoalID = NEW.GoalID;  END; |
| **Prevent Over Capacity in RecyclingCenter** | CREATE OR REPLACE TRIGGER CheckCenterCapacity  BEFORE INSERT OR UPDATE ON Record\_Waste  FOR EACH ROW  DECLARE  total\_waste NUMBER; -- Declare a variable for total waste  center\_capacity NUMBER; -- Declare a variable for center capacity  BEGIN  -- Get the total waste currently recorded for the recycling center  SELECT NVL(SUM(Quantity), 0)  INTO total\_waste  FROM Record\_Waste  WHERE CenterID = :NEW.CenterID;  -- Get the capacity of the recycling center  SELECT Capacity  INTO center\_capacity  FROM RecyclingCenter  WHERE CenterID = :NEW.CenterID;  -- Check if the new total waste exceeds the recycling center's capacity  IF total\_waste + :NEW.Quantity > center\_capacity THEN  RAISE\_APPLICATION\_ERROR(-20002, 'Recycling center capacity exceeded');  END IF;  END;  / |
| **AddRecyclingCenter** | create or replace PROCEDURE AddRecyclingCenter(  p\_center\_name IN VARCHAR2,  p\_center\_location IN VARCHAR2,  p\_center\_contact IN VARCHAR2,  p\_capacity IN NUMBER  )  AS  BEGIN  INSERT INTO RecyclingCenter (CenterName, CenterLocation, CenterContact, Capacity)  VALUES (p\_center\_name, p\_center\_location, p\_center\_contact, p\_capacity);  END;  / |
| **GetWasteByCenter** | create or replace PROCEDURE GetWasteByCenter(  p\_center\_id IN NUMBER  )  AS  BEGIN  -- Use a SELECT INTO if retrieving data into variables, or use an OPEN CURSOR for multiple rows  DBMS\_OUTPUT.PUT\_LINE('Waste records for CenterID: ' || p\_center\_id);  FOR rec IN (  SELECT \*  FROM Record\_Waste  WHERE CenterID = p\_center\_id  ) LOOP  DBMS\_OUTPUT.PUT\_LINE('Record ID: ' || rec.Record\_WasteID ||  ', Quantity: ' || rec.Quantity ||  ', Waste Date: ' || rec.Record\_WasteDate);  END LOOP;  END;  / |
| **UpdateGoalProgress** | create or replace PROCEDURE UpdateGoalProgress(  p\_goal\_id IN NUMBER,  p\_new\_progress IN NUMBER  )  AS  BEGIN  UPDATE SustainabilityGoal  SET GoalProgress = p\_new\_progress  WHERE GoalID = p\_goal\_id;  IF SQL%ROWCOUNT = 0 THEN  RAISE\_APPLICATION\_ERROR(-20001, 'Goal ID not found or no rows updated.');  END IF;  END;  / |
| **AssignWasteToGoal** | create or replace PROCEDURE AssignWasteToGoal(  p\_record\_waste\_id IN NUMBER,  p\_goal\_id IN NUMBER,  p\_target\_reduction IN NUMBER  )  AS  BEGIN  INSERT INTO Waste\_Goal (Record\_WasteID, GoalID, TargetReduction)  VALUES (p\_record\_waste\_id, p\_goal\_id, p\_target\_reduction);  END;  / |
| **ReduceWasteQuantity** | CREATE OR REPLACE PROCEDURE ReduceWasteQuantity(  p\_record\_waste\_id IN NUMBER,  p\_quantity IN NUMBER  )  AS  v\_current\_quantity NUMBER;  BEGIN  -- Fetch the current quantity  SELECT Quantity  INTO v\_current\_quantity  FROM Record\_Waste  WHERE Record\_WasteID = p\_record\_waste\_id;  -- Check if the new quantity would be negative  IF v\_current\_quantity - p\_quantity < 0 THEN  RAISE\_APPLICATION\_ERROR(-20002, 'Waste quantity cannot be negative after recycling.');  ELSE  -- Update the quantity  UPDATE Record\_Waste  SET Quantity = Quantity - p\_quantity  WHERE Record\_WasteID = p\_record\_waste\_id;  END IF;  END; |

**Table 4.6 Waste Reduction & Sustainability Module - Trigger and stored procedure**

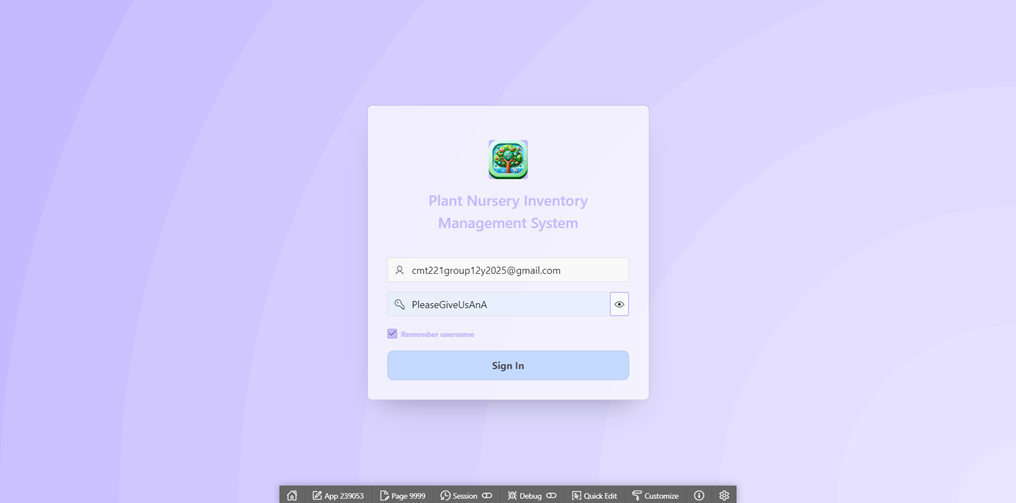
| **TABLE** | **SQL STATEMENT** |
| --- | --- |
| **Activity** | CREATE TABLE Activity (  Activity\_ID NUMBER PRIMARY KEY,  Activity\_Name VARCHAR2(100) NOT NULL,  Activity\_Description VARCHAR2(255)  ); |
| **Plant\_Activity** | CREATE TABLE Plant\_Activity (  Plant\_Activity\_ID NUMBER PRIMARY KEY,  Plant\_ID NUMBER NOT NULL REFERENCES Plant(Plant\_ID) ON DELETE CASCADE ON UPDATE CASCADE,  Activity\_ID NUMBER NOT NULL REFERENCES Activity(Activity\_ID) ON DELETE CASCADE,  Plant\_Activity\_Date DATE NOT NULL,  Plant\_Activity\_Quantity NUMBER NOT NULL  ); |
| **Environmental\_Implication** | CREATE TABLE Environmental\_Implication (  Env\_Imp\_ID NUMBER PRIMARY KEY,  Env\_Imp\_Name VARCHAR2(100) NOT NULL,  Env\_Imp\_Measurement NUMBER NOT NULL,  Env\_Imp\_Threshold NUMBER,  Env\_Imp\_Monitor\_Period DATE  ); |
| **Activity\_Env\_Imp** | CREATE TABLE Activity\_Env\_Imp (  Activity\_Env\_Imp\_ID NUMBER PRIMARY KEY,  Activity\_ID NUMBER NOT NULL REFERENCES Activity(Activity\_ID) ON DELETE CASCADE,  Env\_Imp\_ID NUMBER NOT NULL REFERENCES Environmental\_Implication(Env\_Imp\_ID) ON DELETE CASCADE  ); |
| **Analysis\_Report** | CREATE TABLE Analysis\_Report (  Analysis\_Report\_ID NUMBER PRIMARY KEY,  Analysis\_Report\_Date DATE DEFAULT SYSDATE NOT NULL,  Analysis\_Report\_Summary VARCHAR2(4000)  ); |
| **Analysis\_Env\_Imp** | CREATE TABLE Analysis\_Env\_Imp (  Analysis\_Env\_Imp\_ID NUMBER PRIMARY KEY,  Analysis\_Report\_ID NUMBER NOT NULL REFERENCES Analysis\_Report(Analysis\_Report\_ID) ON DELETE CASCADE,  Env\_Imp\_ID NUMBER NOT NULL REFERENCES Environmental\_Implication(Env\_Imp\_ID) ON DELETE CASCADE  ); |

**Table 4.7 Environmental Impact Module**

| **SEQUENCE** | **SQL STATEMENT** |
| --- | --- |
| **Sequence for Environmental\_Implication** | CREATE SEQUENCE Environmental\_Implication\_SEQ START WITH 1 INCREMENT BY 1; |
| **Sequence for Activity** | CREATE SEQUENCE Activity\_SEQ START WITH 1 INCREMENT BY 1; |
| **Sequence for Activity\_Env\_Imp** | CREATE SEQUENCE Activity\_Env\_Imp\_SEQ START WITH 1 INCREMENT BY 1; |
| **Sequence for Analysis\_Report** | CREATE SEQUENCE Analysis\_Report\_SEQ START WITH 1 INCREMENT BY 1; |
| **Sequence for Analysis\_Env\_Imp** | CREATE SEQUENCE Analysis\_Env\_Imp\_SEQ START WITH 1 INCREMENT BY 1; |
| **Sequence for Plant\_Activity** | CREATE SEQUENCE Plant\_Activity\_SEQ START WITH 1 INCREMENT BY 1; |

| **TRIGGER** | **SQL STATEMENT** |
| --- | --- |
| **Trigger for Environmental\_Implication** | CREATE OR REPLACE TRIGGER trg\_env\_imp\_id  BEFORE INSERT ON Environmental\_Implication  FOR EACH ROW  BEGIN  :NEW.Env\_Imp\_ID := Environmental\_Implication\_SEQ.NEXTVAL;  END;  / |
| **Trigger for Activity** | CREATE OR REPLACE TRIGGER trg\_activity\_id  BEFORE INSERT ON Activity  FOR EACH ROW  BEGIN  :NEW.Activity\_ID := Activity\_SEQ.NEXTVAL;  END;  / |
| **Trigger for Activity\_Env\_Imp** | CREATE OR REPLACE TRIGGER trg\_activity\_env\_imp\_id  BEFORE INSERT ON Activity\_Env\_Imp  FOR EACH ROW  BEGIN  :NEW.Activity\_Env\_Imp\_ID := Activity\_Env\_Imp\_SEQ.NEXTVAL;  END;  / |
| **Trigger for Analysis\_Report** | CREATE OR REPLACE TRIGGER trg\_analysis\_report\_id  BEFORE INSERT ON Analysis\_Report  FOR EACH ROW  BEGIN  :NEW.Analysis\_Report\_ID := Analysis\_Report\_SEQ.NEXTVAL;  END;  / |
| **Trigger for Analysis\_Env\_Imp** | CREATE OR REPLACE TRIGGER trg\_analysis\_env\_imp\_id  BEFORE INSERT ON Analysis\_Env\_Imp  FOR EACH ROW  BEGIN  :NEW.Analysis\_Env\_Imp\_ID := Analysis\_Env\_Imp\_SEQ.NEXTVAL;  END;  / |
| **Trigger for Plant\_Activity** | CREATE OR REPLACE TRIGGER trg\_plant\_activity\_id  BEFORE INSERT ON Plant\_Activity  FOR EACH ROW  BEGIN  :NEW.Plant\_Activity\_ID := Plant\_Activity\_SEQ.NEXTVAL;  END;  / |

**Table 4.8 Environmental Impact Module - Trigger and stored procedure**

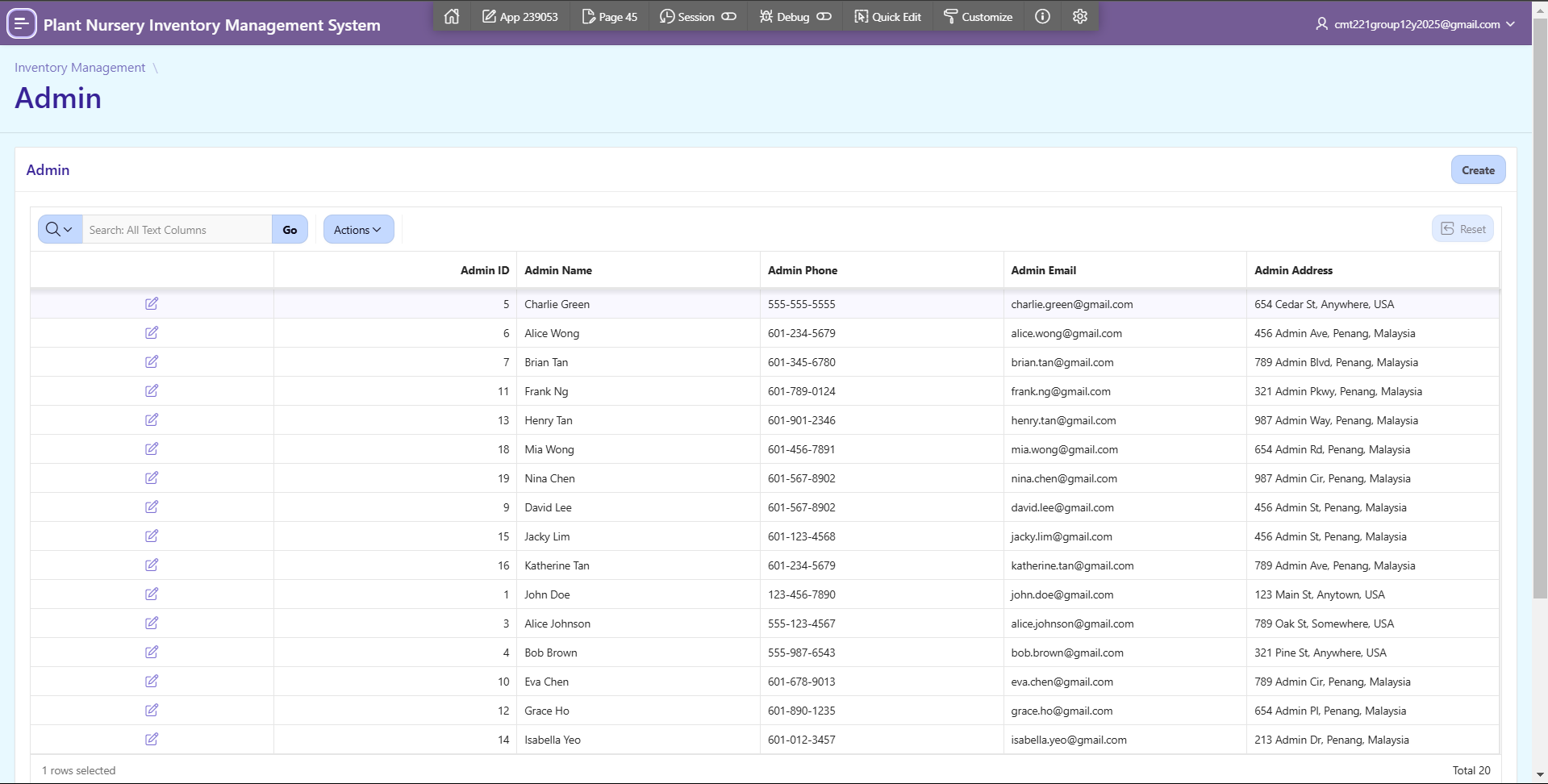


**Figure 5.8 Plant Nursery Inventory Management Login Page**

When we start the system, the login page is displayed. The admin needs to input the correct username and password to login into the Plant Nursery Inventory Management system. This is to ensure only the admin has access to the database system. The username for the login page is cmt221group12y2025@gmail.com while the password is PleaseGiveUsAnA.

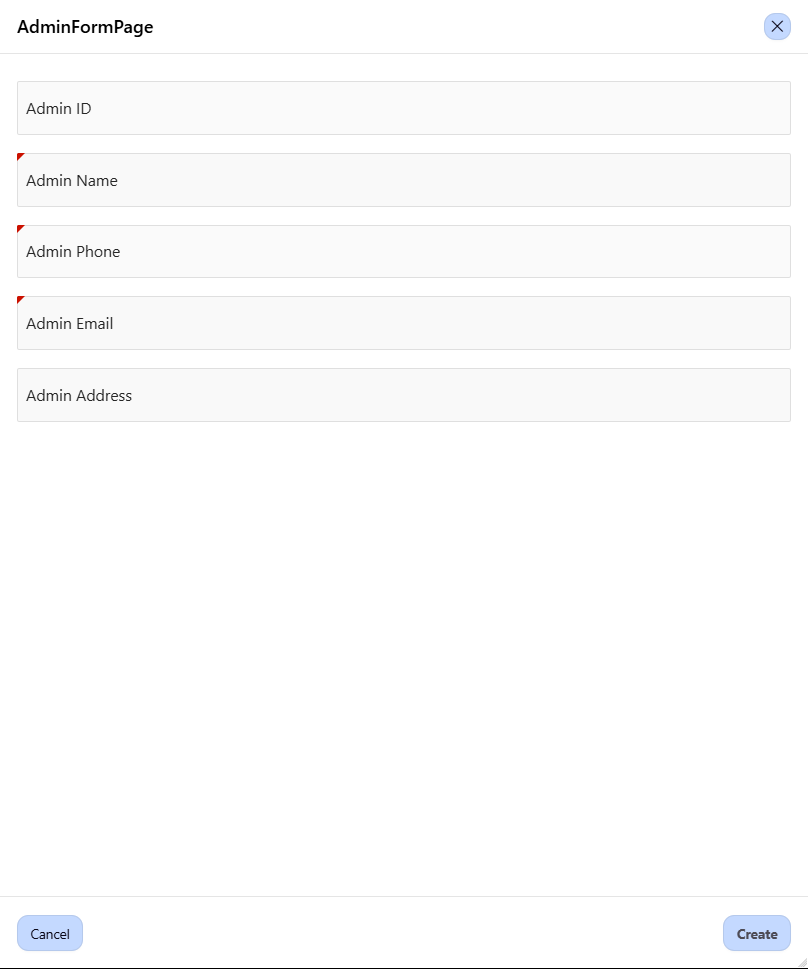


**Figure 5.9 Plant Nursery Inventory Management Menu**



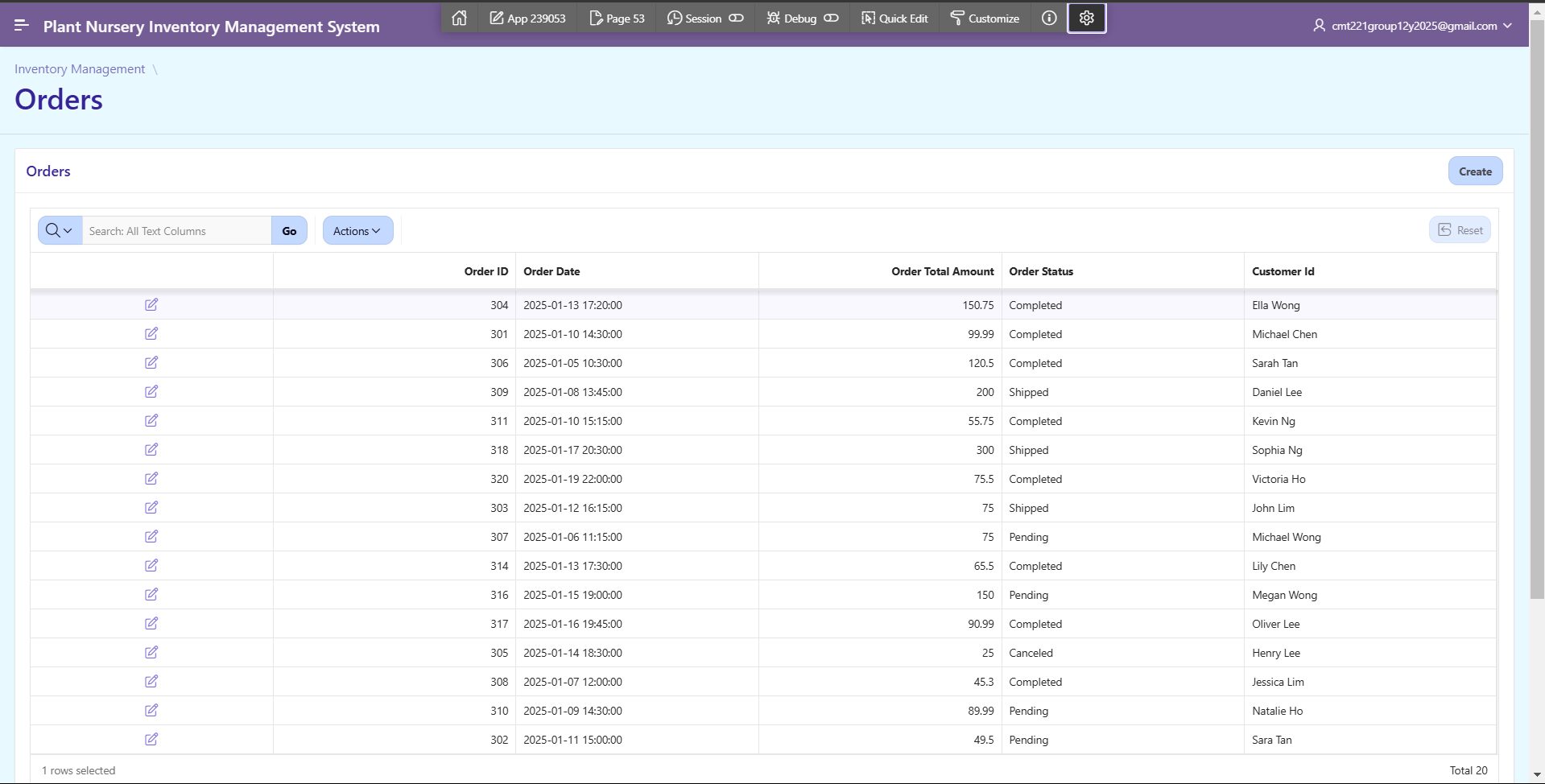
**Figure 5.10 Inventory Management Module - Admin page**

This is the page about the admin information. The admin can see the other admin information, but for the website they will use the cmt221group12y2025@gmail.com to login.



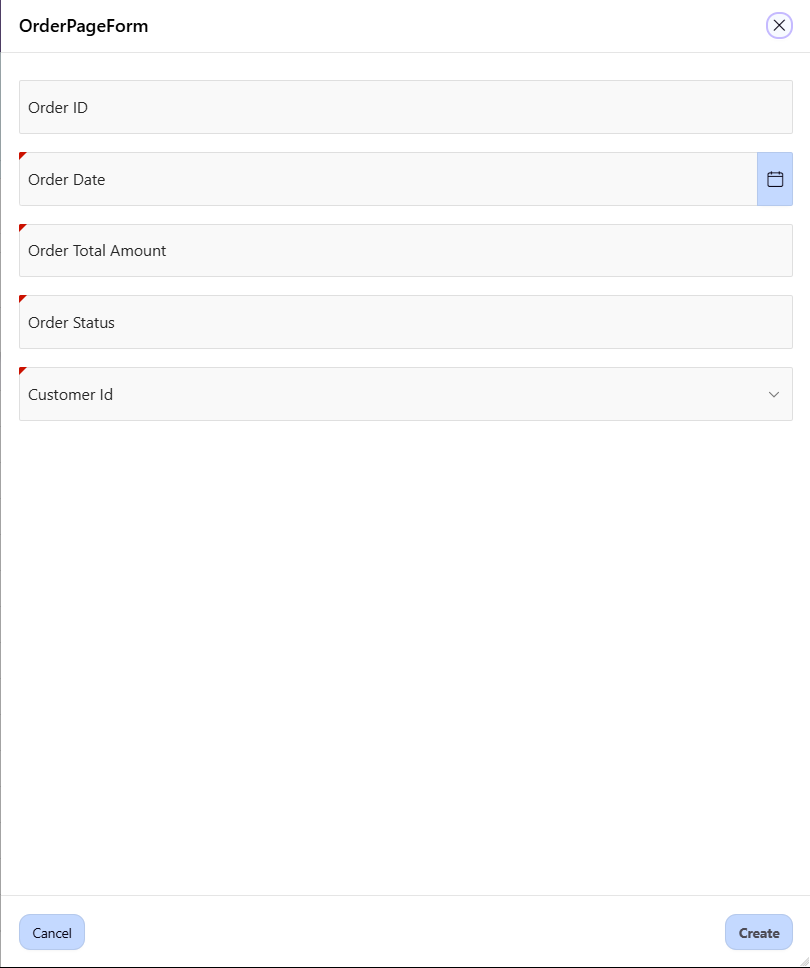
**Figure 5.11 Inventory Management Module - Admin Form page**

This form page allows the admin to add others or new admin information which includes ID, name, phone, email and address.



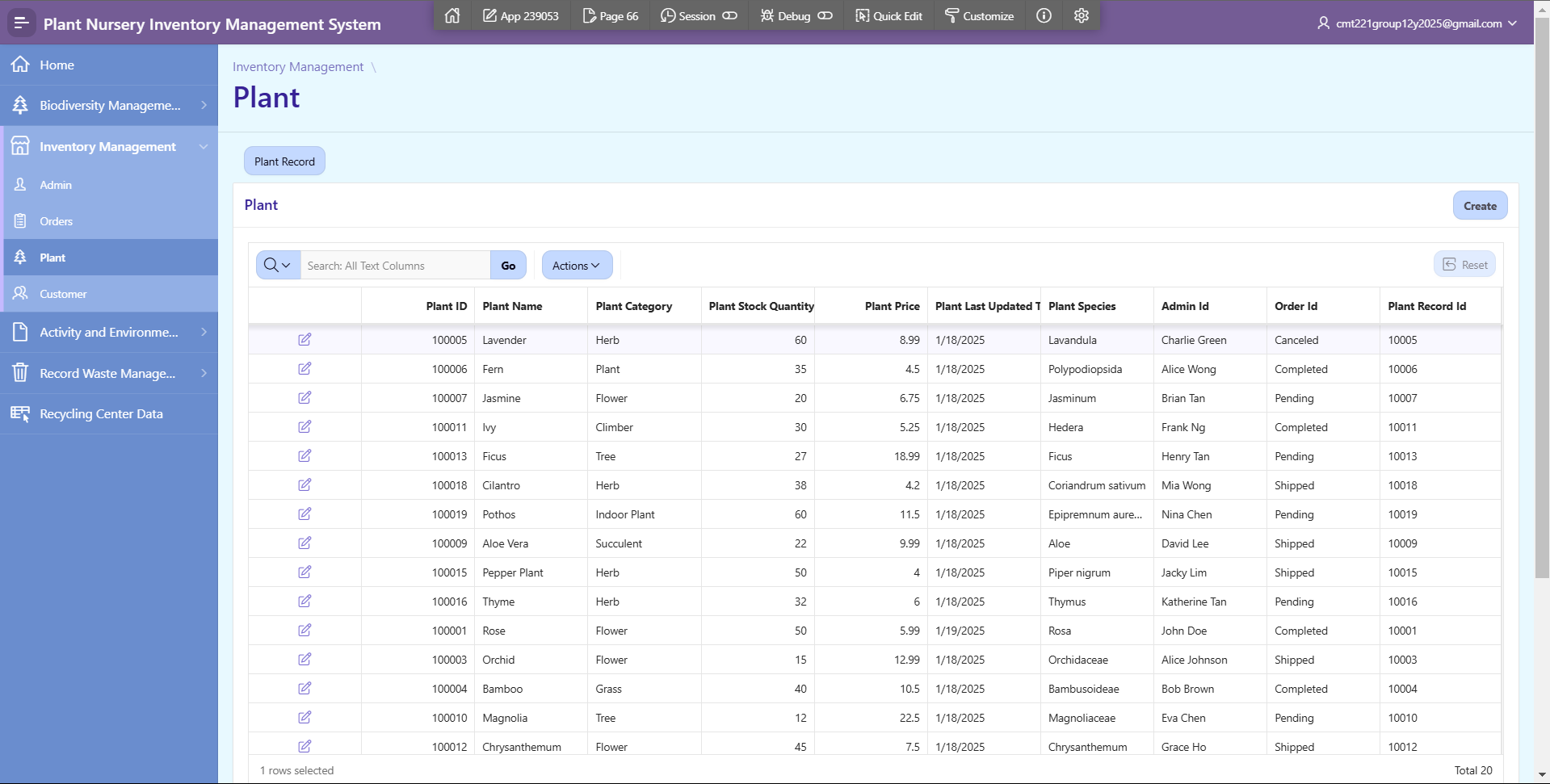
**Figure 5.12 Inventory Management Module - Orders page**

This is the page about the orders information. The admin can see the orders information like order id, order date, order total amount and customer id.



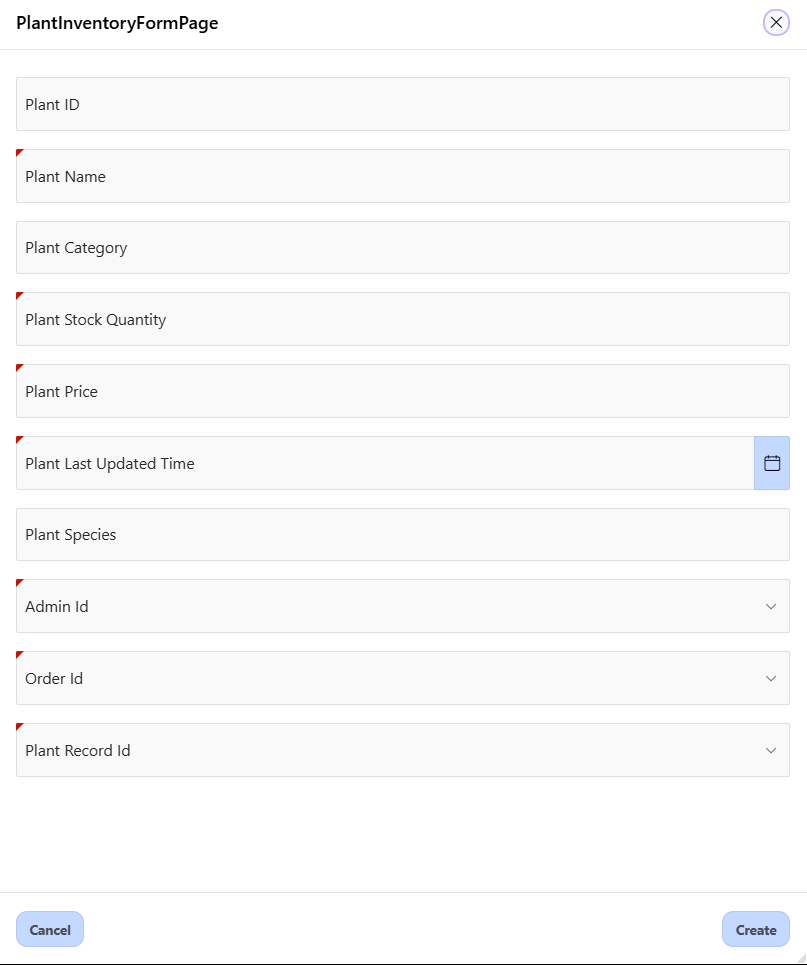
**Figure 5.13 Inventory Management Module - Orders form page**

This form page allows the admin to edit others or new orders information which includes ID, date, total amount, status and customer id too.



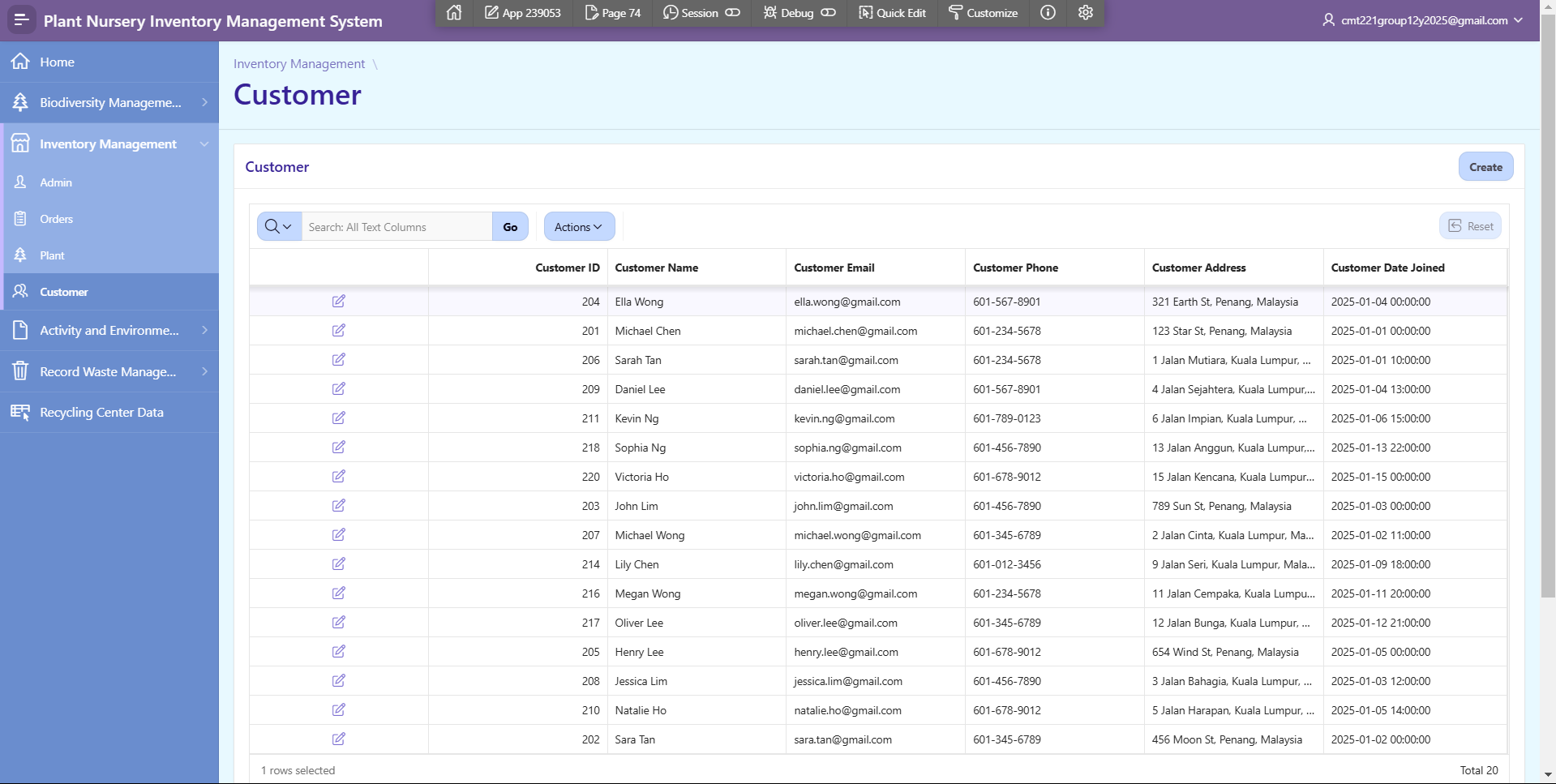
**Figure 5.14 Inventory Management Module - Plant page**

This is the page about the plant information. The admin can see the plant information like plant id, name, category, stock quantity, last updated date, species, admin id, order id and plant record id.



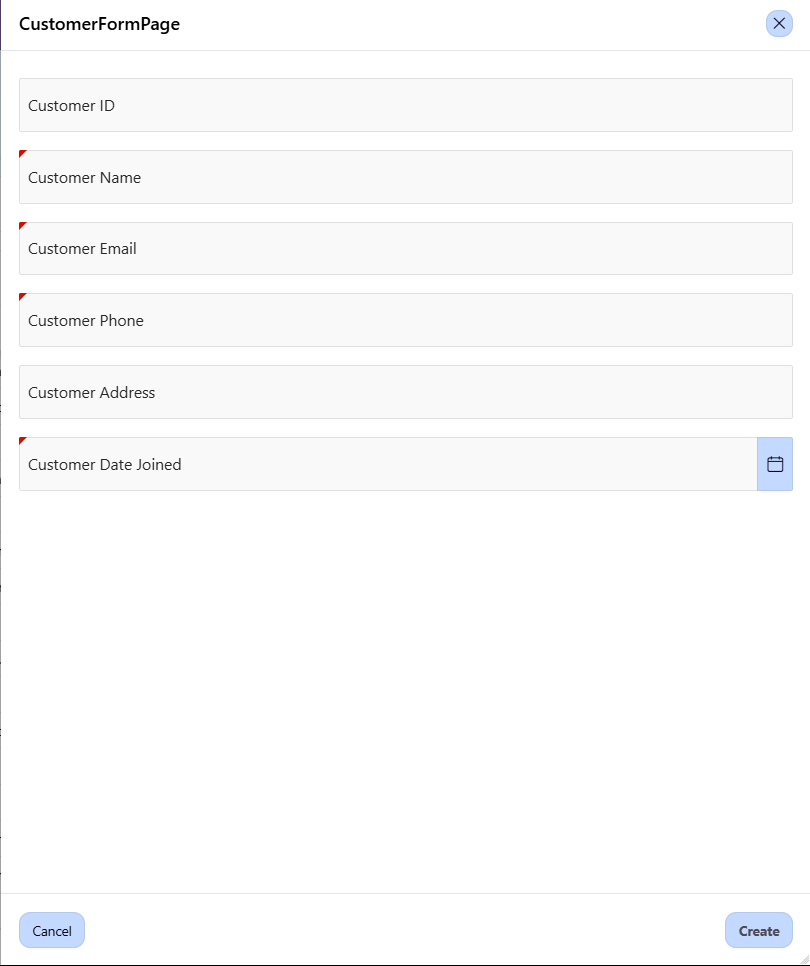
**Figure 5.15 Inventory Management Module - Plant form page**

This form page allows the admin to edit others or new plant information which includes plant id, name, category, stock quantity, last updated date, species, admin id, order id and plant record id.



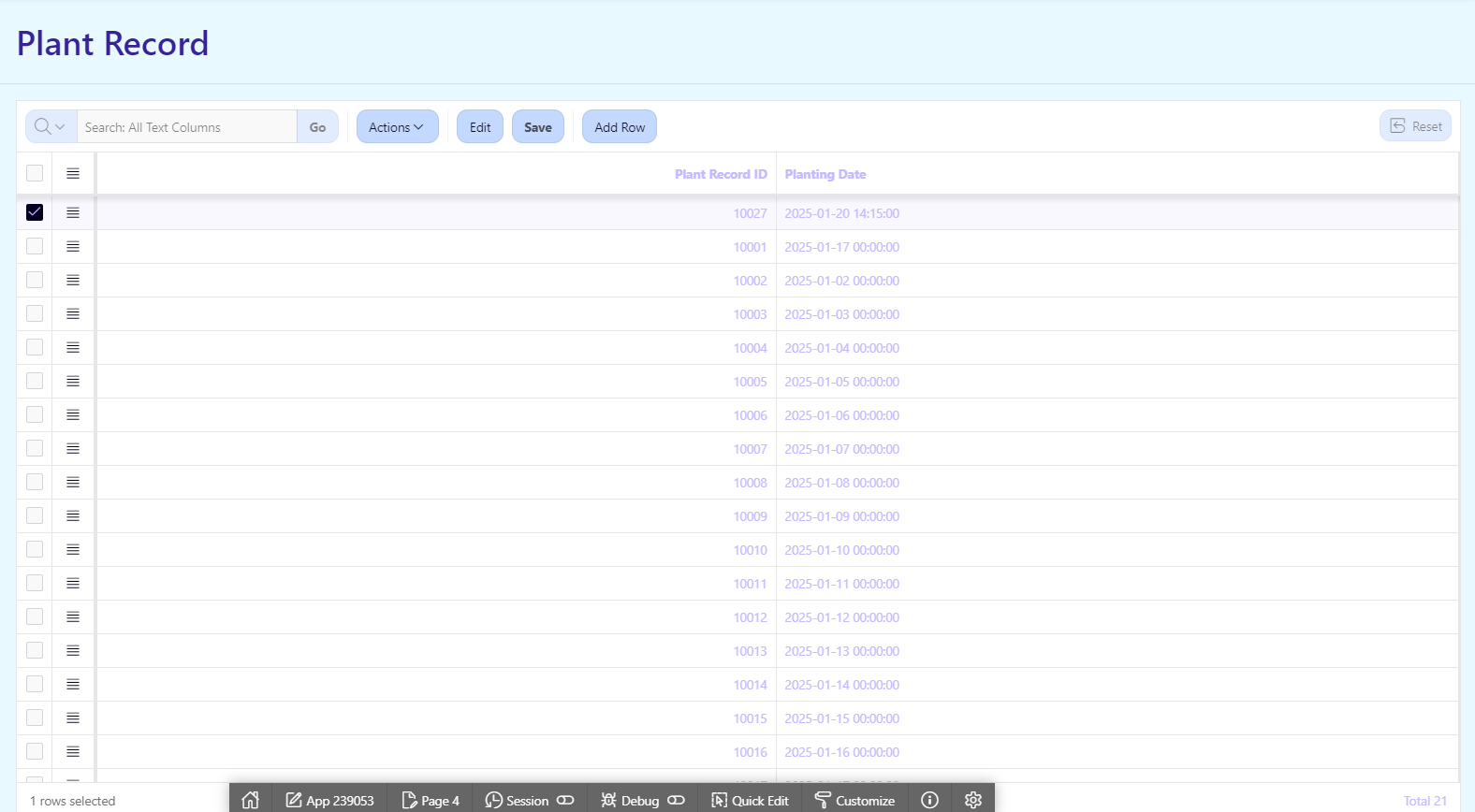
**Figure 5.16 Inventory Management Module - Customer page**

This form page allows the admin to edit others or new customer information which includes customer id, name, email, phone, address and date joined.



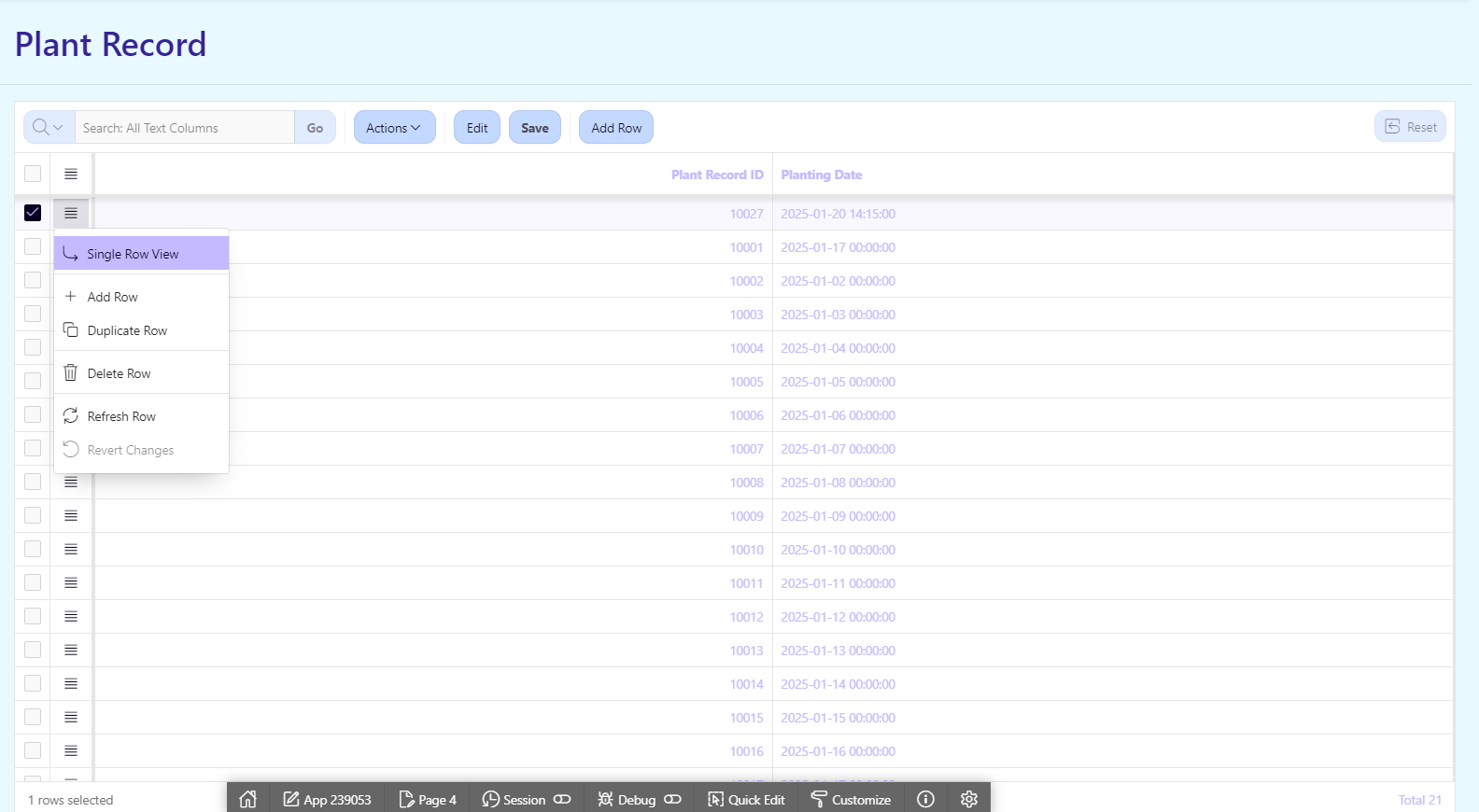
**Figure 5.17 Inventory Management Module - Customer form page**

This form page allows the admin to edit others or new customer information which includes customer id, name, email, phone, address and date joined.



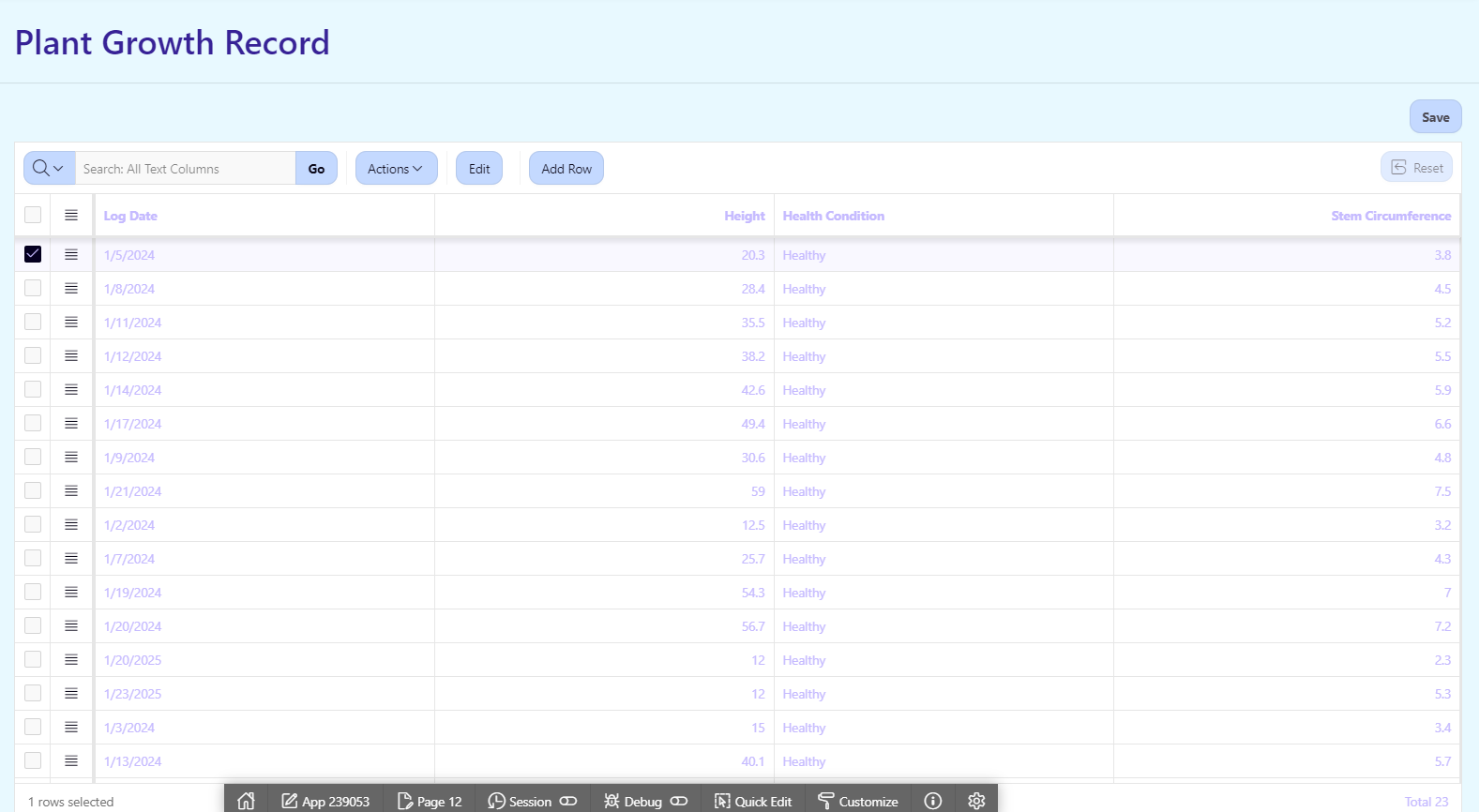
*Figure 5.18 Plant Record Page*

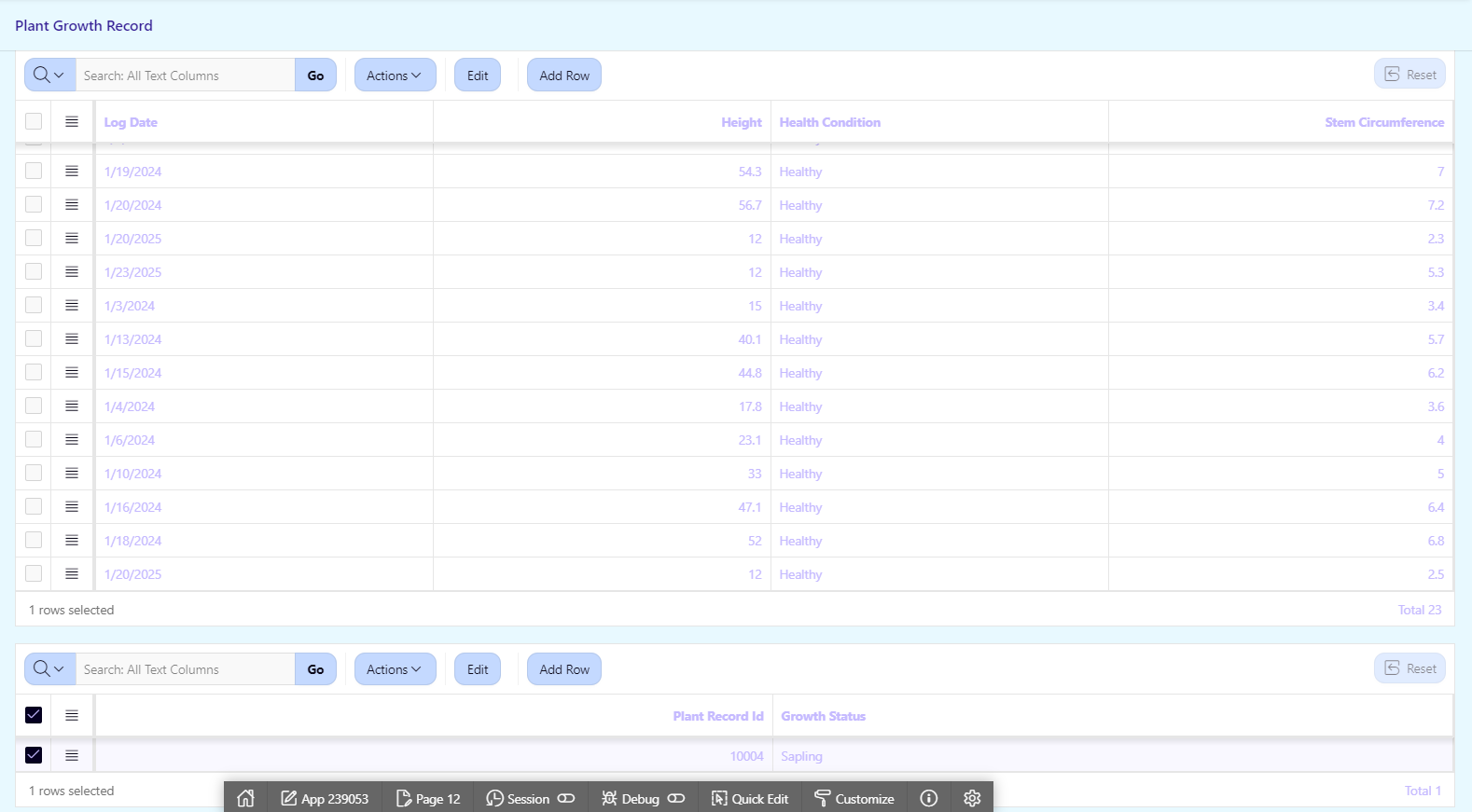
This is the Plant Record Report Page. The Plant Record ID and Planting Date are all displayed on this page. The admin will be able to edit and delete existing Plant Record ID’s and Planting Date’s as well as add new ones. The admin will also be able to search for data from the table.

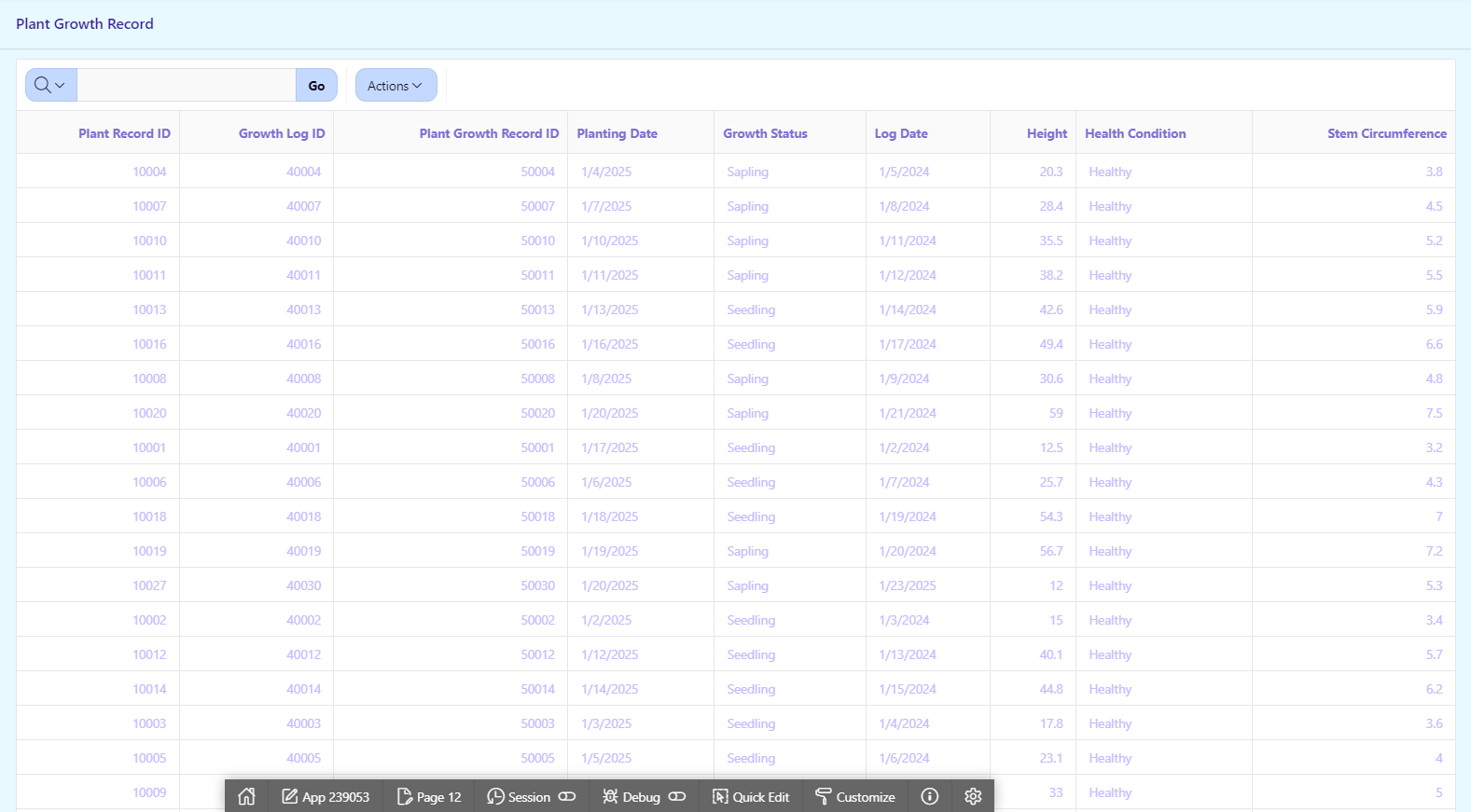


*Figure 5.19 Plant Record Functionalities*

The functionalities for Plant Record Report Page can be accessed by clicking the hamburger icon next to each row. It can also be accessed by using the ‘Edit’, ‘Save’ and ‘Add Row’ buttons located below the Plant Record title and above the data.

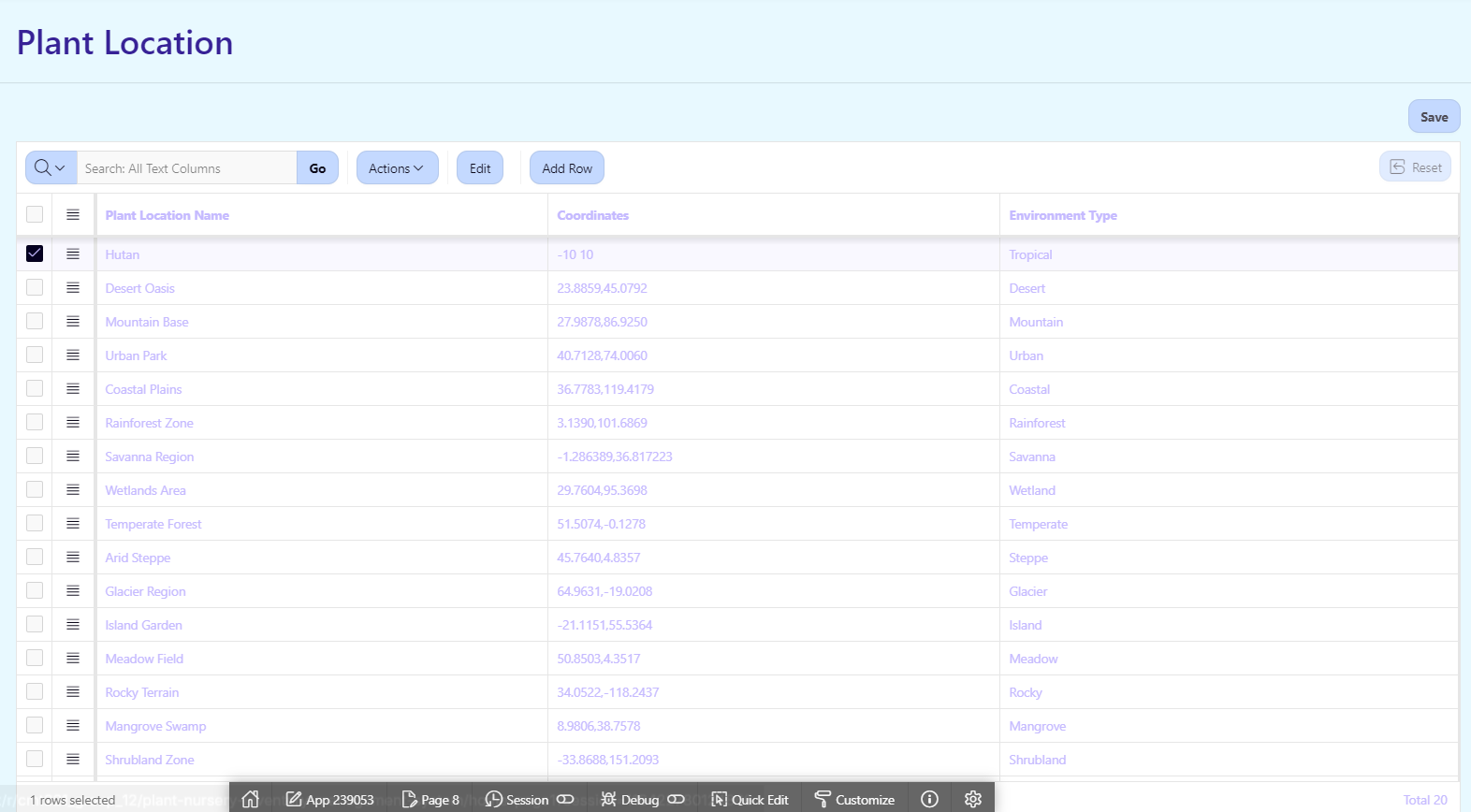


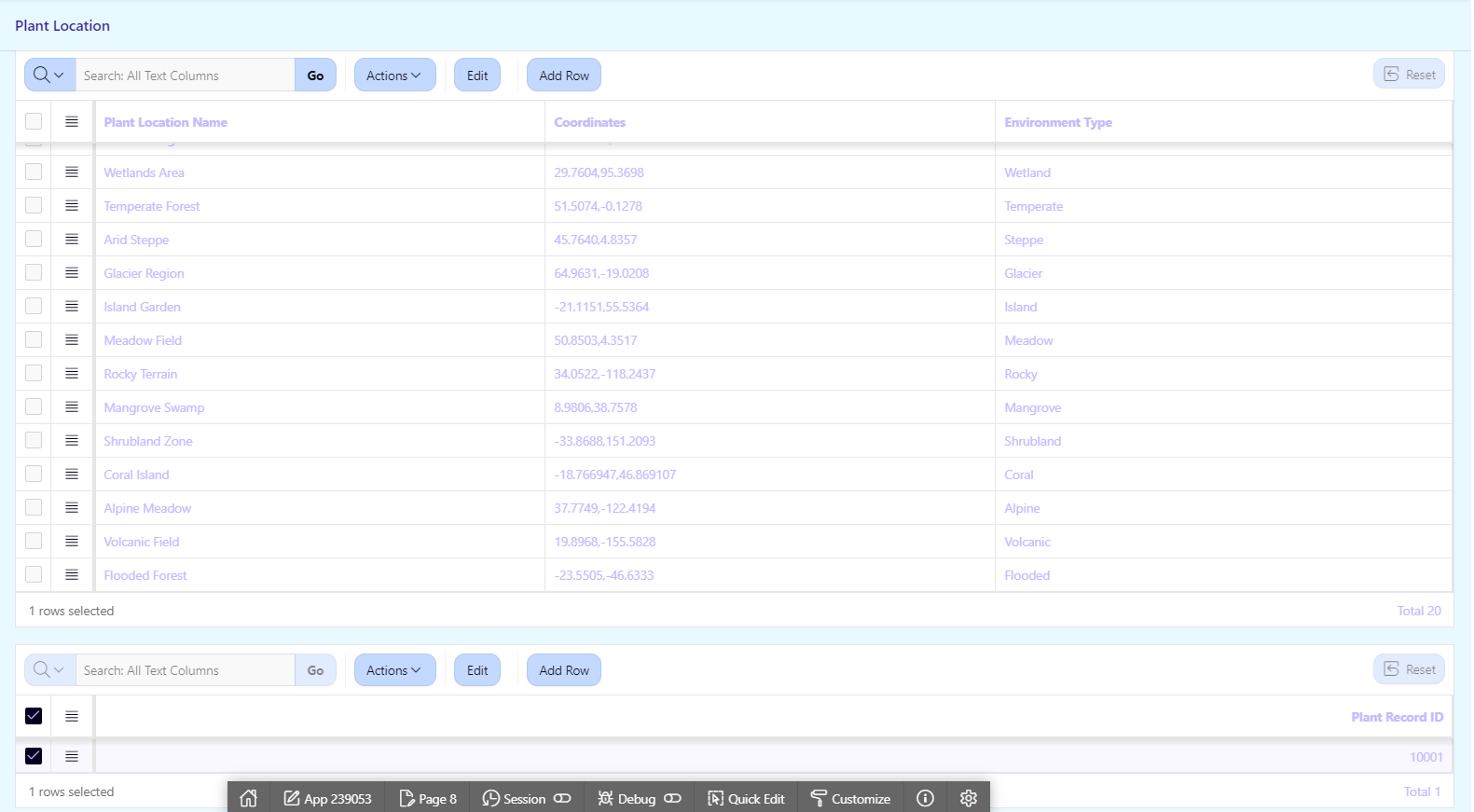




*Figure 5.20 Plant Growth Record Report Page*

This is the Plant Growth Record Report Page. The Log Date, Height, Health Condition, Stem Circumference, Plant Record ID and Growth Status is displayed on this page. The admin will also be able to edit and delete existing data as well as add new ones. The existing data will also be able to be seen and searched at the bottom of the page for quick access by the admin.

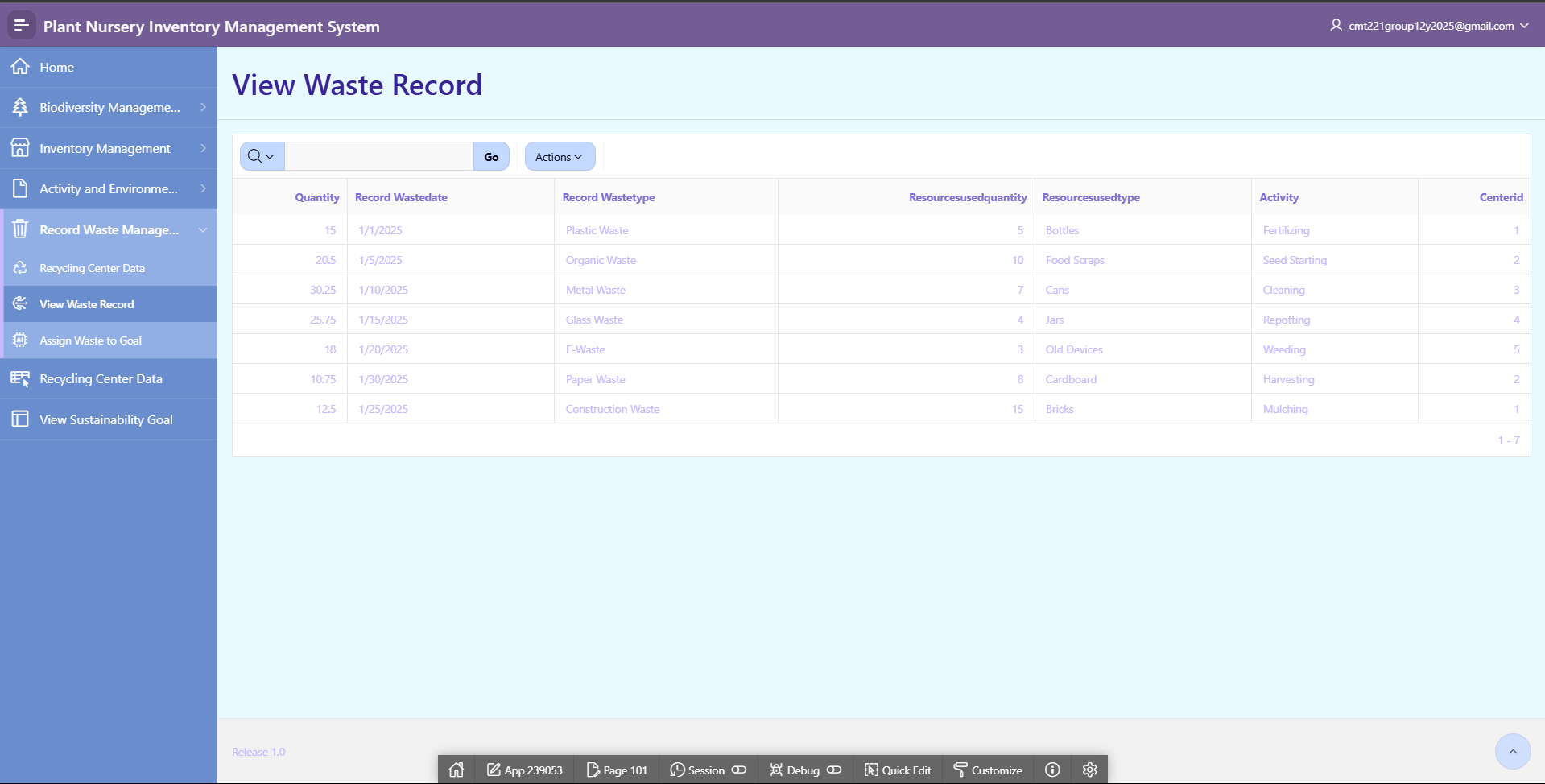




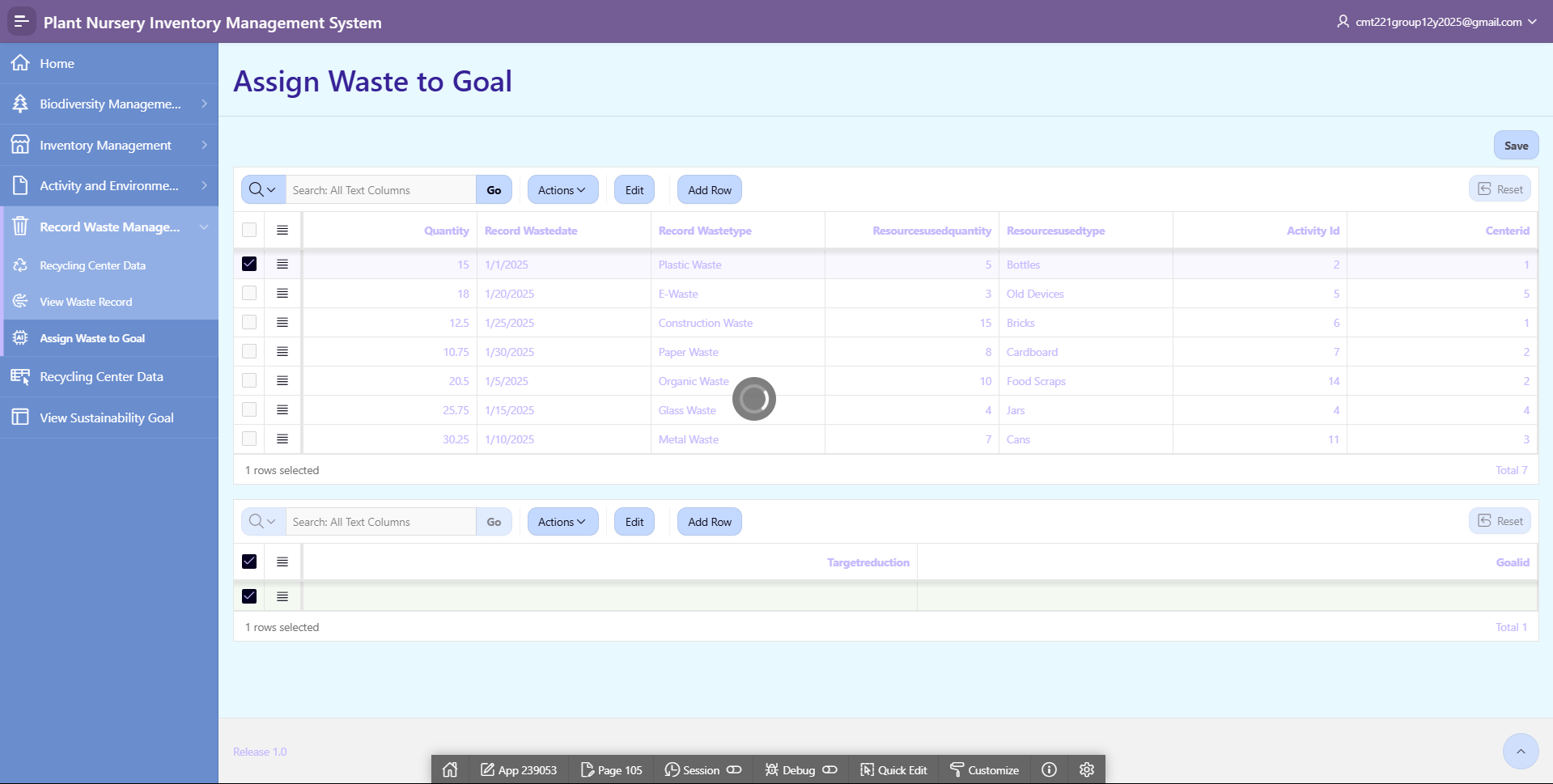


*Figure 5.21 Plant Location Report Page*

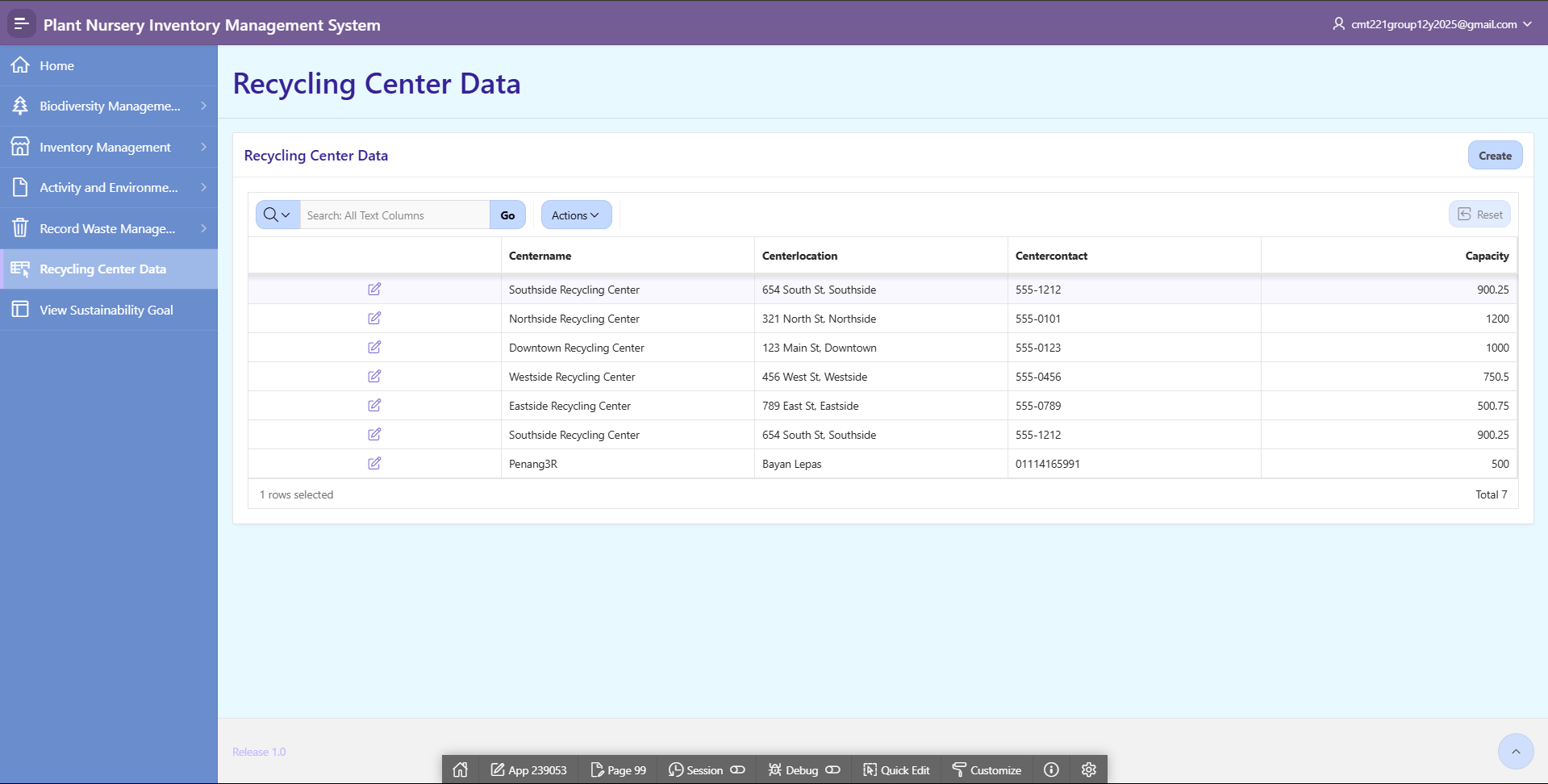
This is the Plant Location Report Page. The Plant Location Name, Coordinates, Environment Type and Plant Record ID is displayed on this page. The admin will also be able to edit and delete existing data as well as add new ones. The existing data will also be able to be seen and searched at the bottom of the page for quick access by the admin.



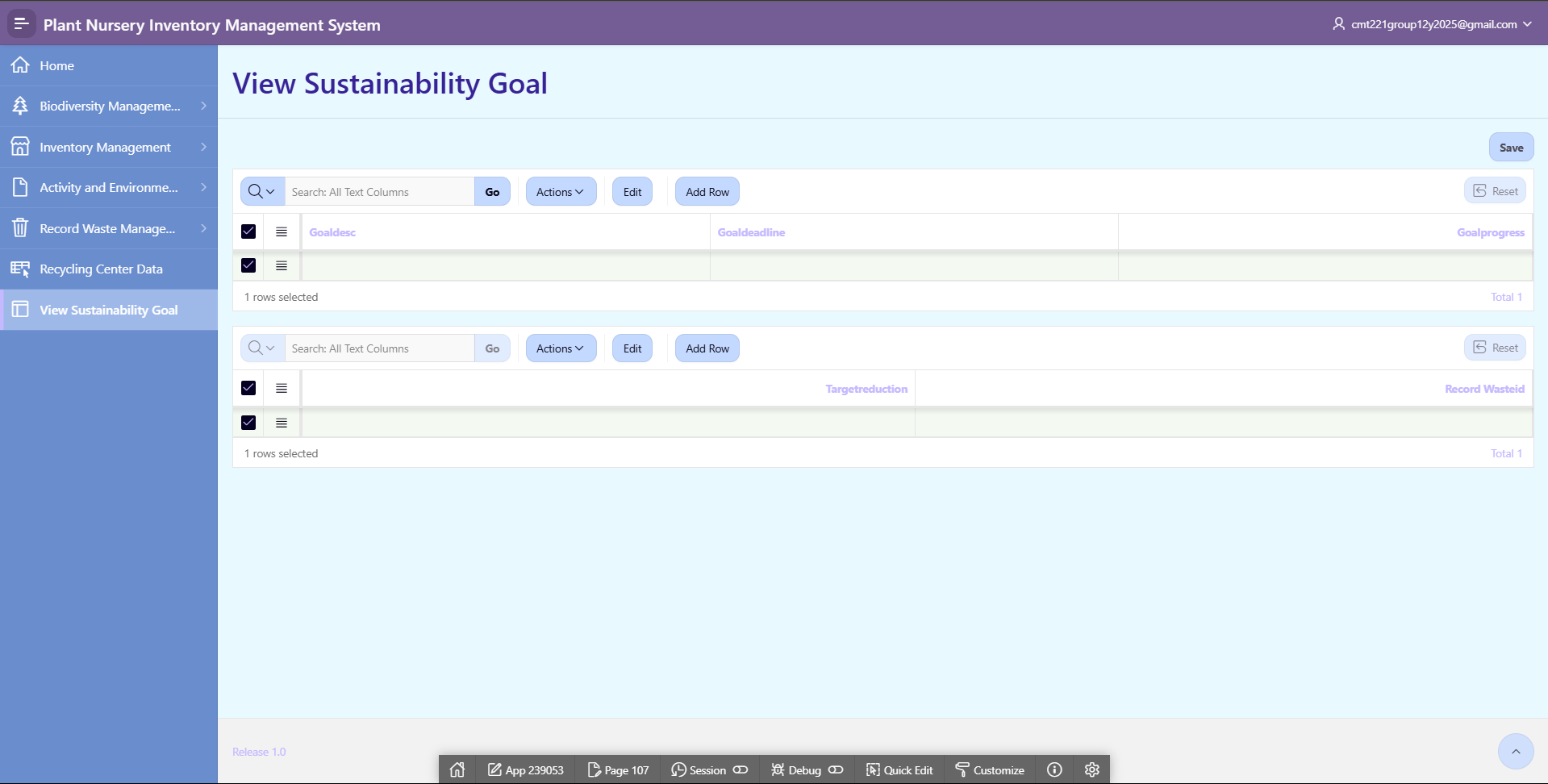
*Figure 5.22 Waste Record Module Table Page*



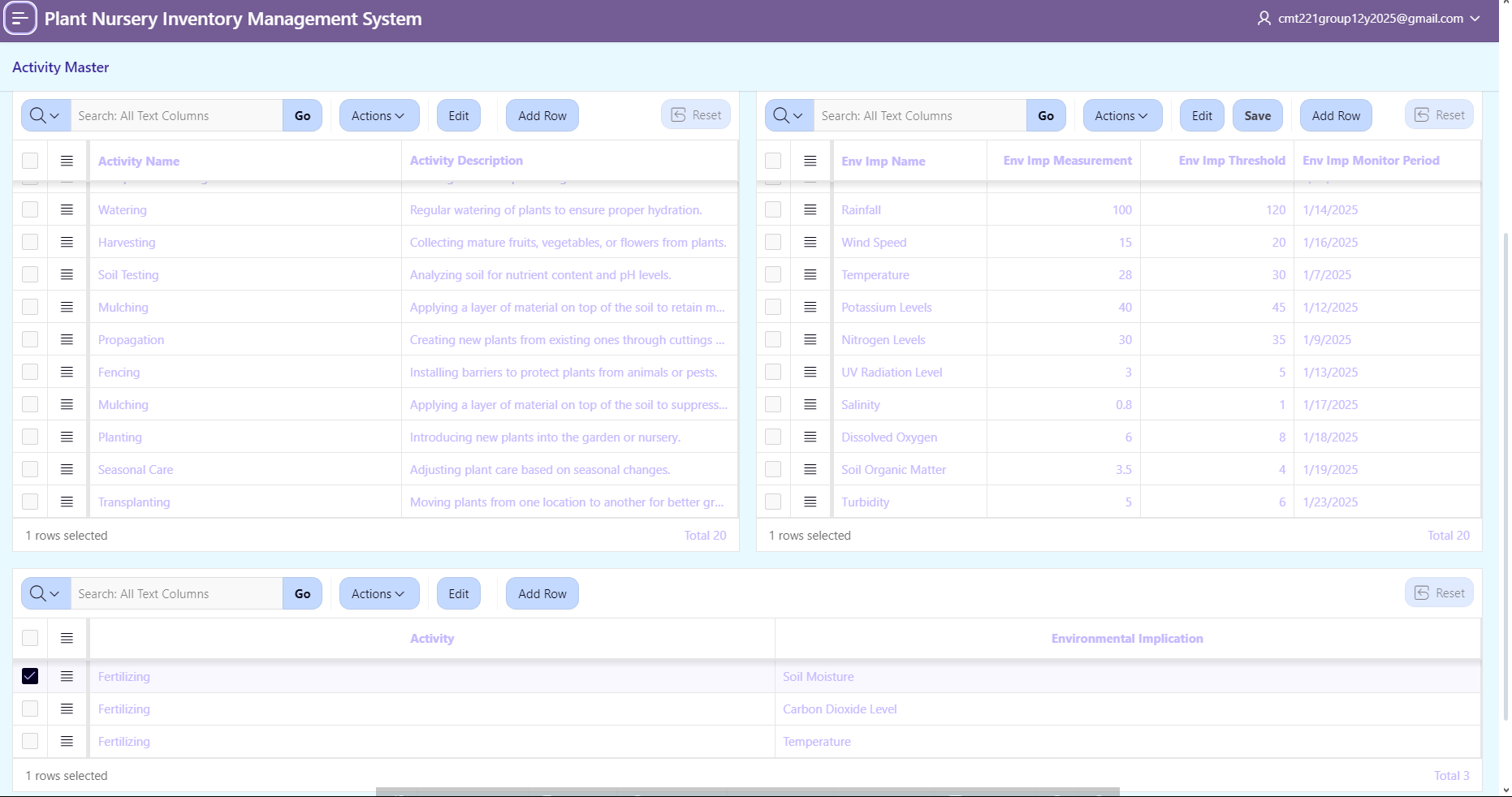
*Figure 5.23 Waste Record Module Table Page*



*Figure 5.24 Waste Record Module Table Page*

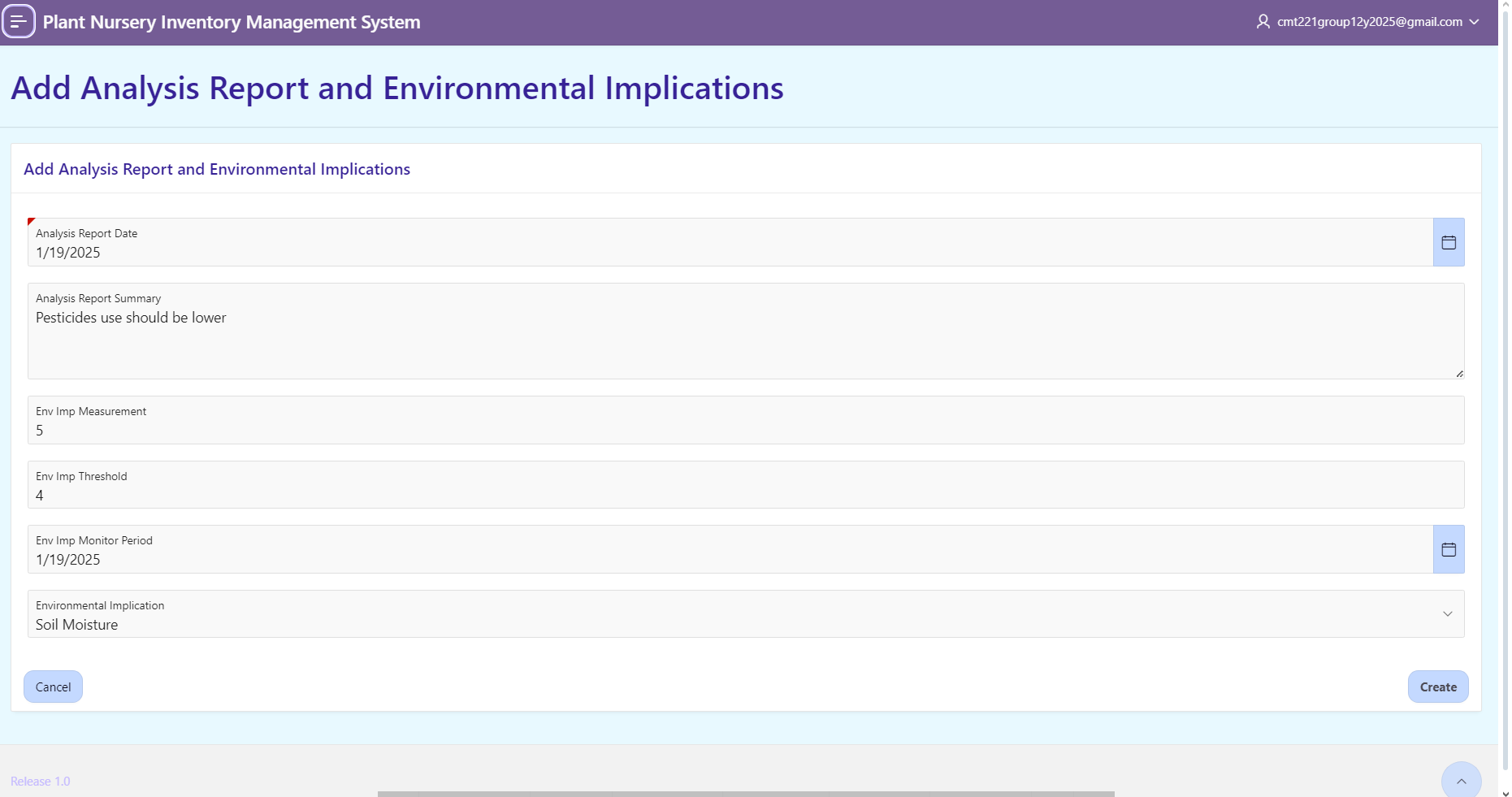


*Figure 5.25 Waste Record Module Table Page*



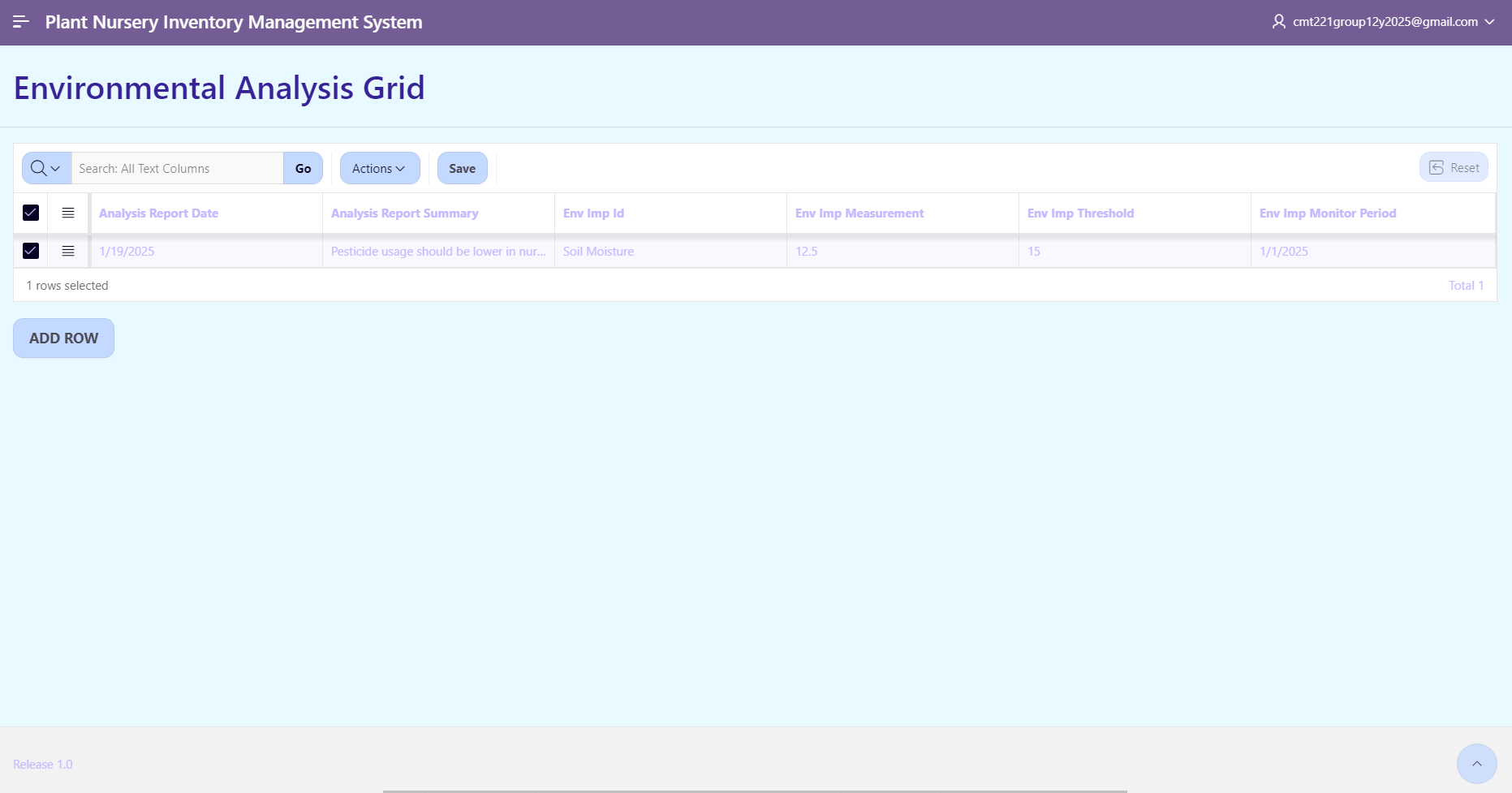
*Figure 5.26 Activity Environmental Implication Table Page*

This is the Activity Environmental Implication Table Page. The Activity Name, and Description shown in the left table and in the right table has the Environmental Implication Name, Measurement, Threshold, and Monitor Period displayed. The admin is able to click one of the activity names and can check what are the Environmental Implications caused by the activity. Admin is also able to add new data by clicking on an activity name and selecting the environmental implication name from list.



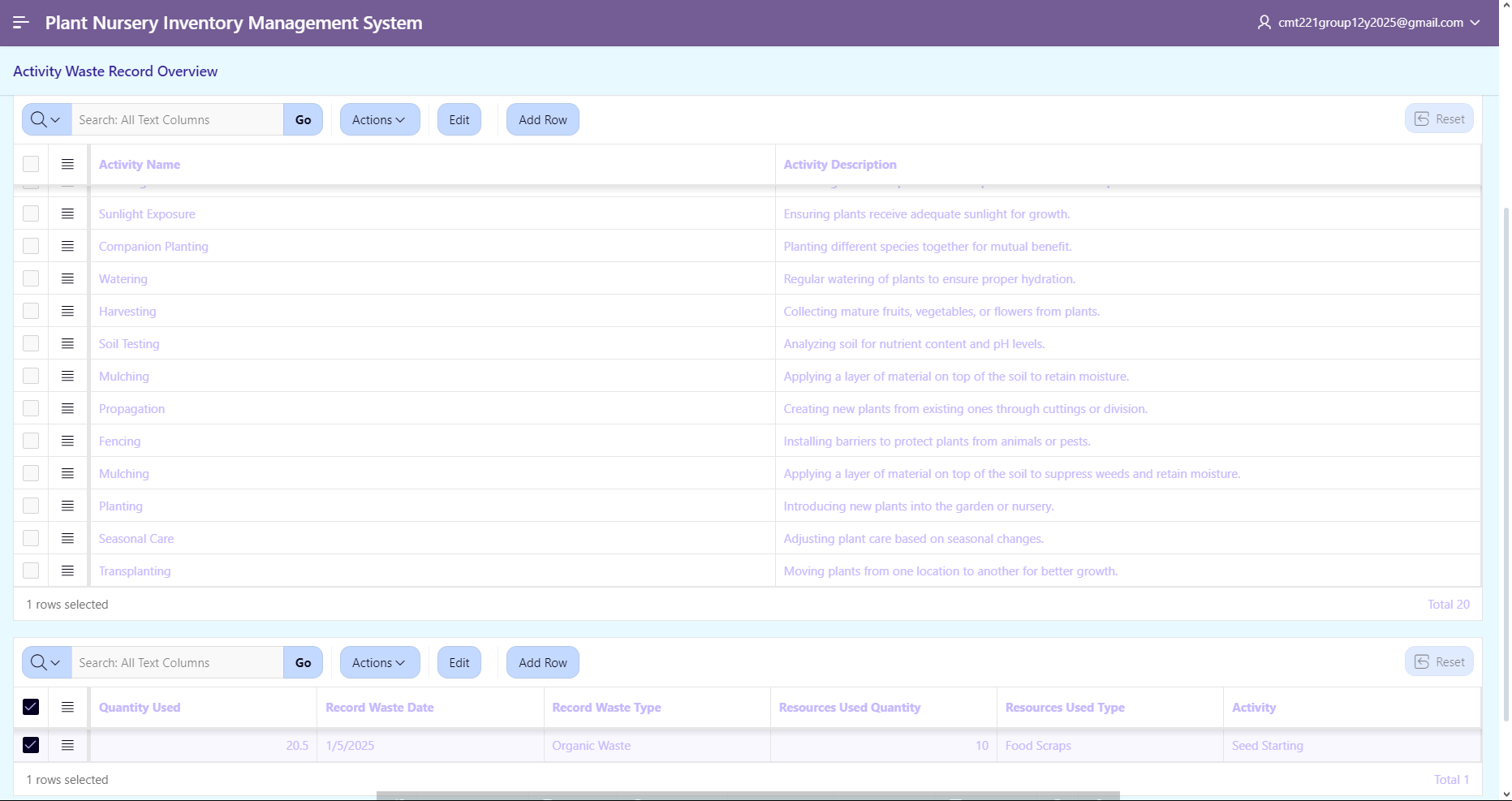
*Figure 5.27 Add Analysis Report and Environmental Implication Table Page*

In this page, administrators can type in the data they want to insert such as Analysis Report Data, Analysis Report Summary, Environmental Implication Name, Measurement, Threshold, and Monitor Period. After clicking the ‘Create’ button, the page will redirect to another page in this application which is ‘Environmental Analysis Grid’ page and data will be saved there.



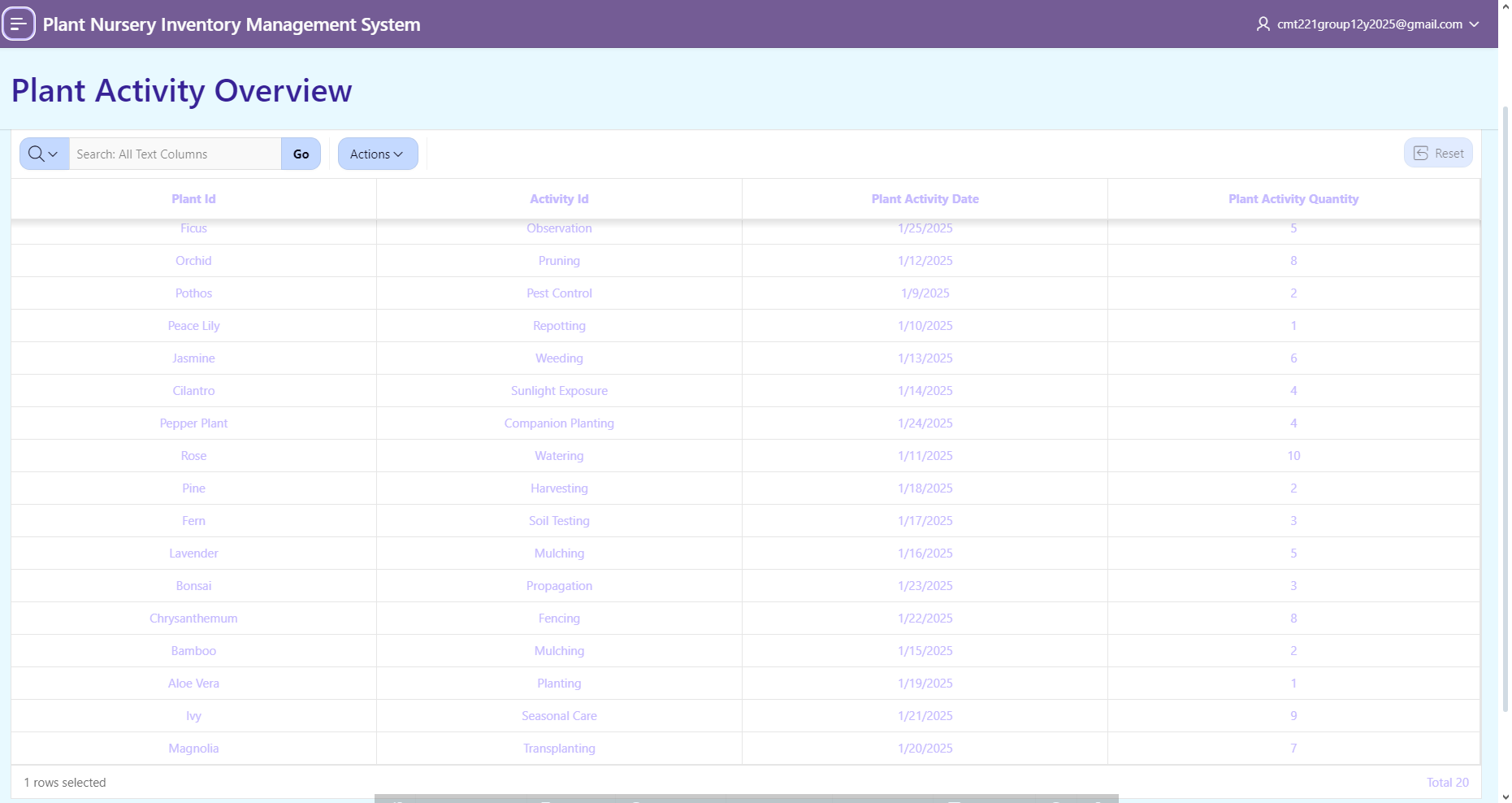
*Figure 5.28 Environmental Analysis Grid Page*

This page shows all the rows inserted from the ‘Add Analysis Report and Environmental Implication Table’ page. In this page, the inserted row is displayed with columns of Analysis Report Date, Analysis Report Summary, Environmental Implication ID, Measurement, Threshold, and Monitor Period.



*Figure 5.29 Activity Waste Record Page*

This page consists of two tables which are the Activity table on the top and the Waste Record table with. In the Activity table, there are Activity Name and Activity Description columns and in the Waste record table, there are Activity ID, Waste Record Date, Waste Record Type, Resource Quantity, Used Quantity, and Resource Type. When the admin clicks any of the activity rows, it will show the respective row with the Waste Record.



*Figure 5.30 Plant Activity Overview Page*

This page shows the data of Plant and Activity they both are related to. This table has columns like Plant ID, Activity ID, Plant Activity Date, and Plant Activity Quantity.