LONDON
METROPOLITAN
UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CS5002NI SOFTWARE ENGINEERING

Assessment Weightage & Type

35% Individual Coursework

Year and Semester

2022-23 Spring

Student Name:Apekshya Adhikari

London Met ID:22085762

College ID: NP01AI4S230005

Assignment Due Date:May 07,2024

Assignment Submission Date:May 07,2024

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.

Table of Contents

Table of Tables

Table of Figures

1.Introduction:

In the heart of Lalitpur,Nepal located in the scenic town of Godawari, lies the McGregor Institute of Botanical Training.For nearly seven years,McGregor has been a beacon of knowledge and passion for those with a deep rooted interest in agriculture and horticulture. Affiliated with Dublin City University,McGregor has been a steady provider of undergraduate and postgraduate courses nurturing budding talents and cultivating expertise in the green reals of agriculture and horticulture.

With an unpredectatable surge in individuals seeking to engage into the world of horticulture,McGregor is introducing a series of short term certificate courses.These courses will not only impart knowledge but also ignite a passion for cultivating and nurturing the wonders around us.McGregor plans to offer a diverse array of plant varieties and other freely distributed demonstrating access to nature's bounty.

Central to McGregor's mission is the creation of dynamic platform a forum where individuals from all over the world can converge to share their passion,exchange ideas and spearhead initiatives aimed at conserving rare plants and safeguarding our forests.This forum will serve as a melting pot of knoweldge and expertise where novices can seek guidance from seasoned experts,and where the collective wisdom can flourish.McGregor is here to help you grow your Green Thumb and make a difference in the world.

2.WBS(Work Breakdown Structure):

Breaking work into smaller tasks is a common productivity technique used to make the work more manageable and approachable. For projects, the **Work Breakdown Structure (WBS)** is the tool that utilizes this technique and is one of the most important project management documents. It singlehandedly integrates scope, cost and schedule baselines ensuring that project plans are in alignment.

The Project Management Institute (PMI) Project Management Book of Knowledge (PMBOK) defines the Work Breakdown Structure as a “deliverable oriented hierarchical decomposition of the work to be executed by the project team.
(WorkBreakdownStructure, 2023)

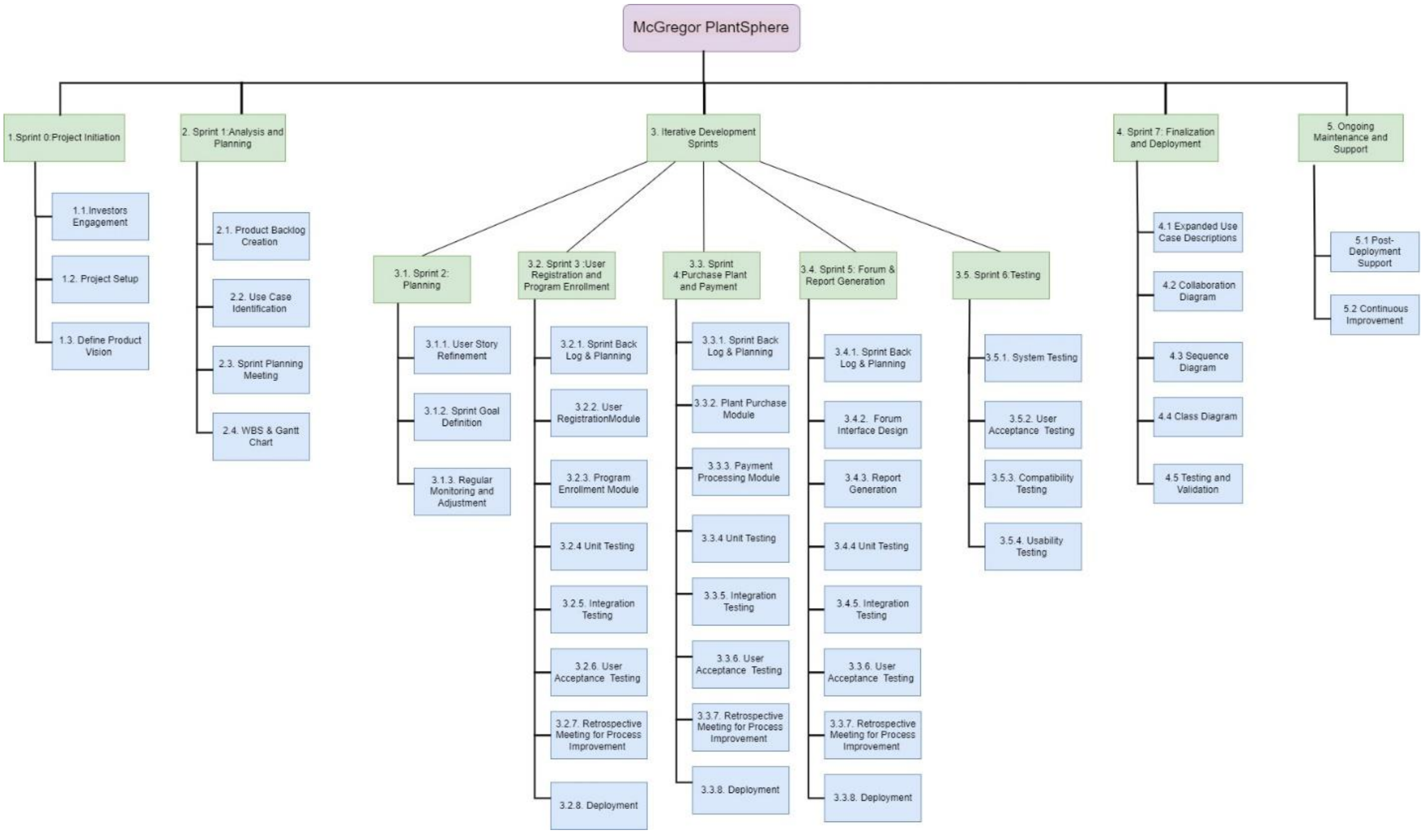

McGregor PlantSphere
1.Sprint 0:Project Initiation
1.1.Investors Engagement
1.2. Project Setup
1.3. Define Product Vision
2. Sprint 1:Analysis and Planning
2.1. Product Backlog Creation
2.2. Use Case Identification
2.3. Sprint Planning Meeting
2.4. WBS & Gantt Chart
3. Iterative Development Sprints
3.1. Sprint 2: Planning
3.1.1. User Story Refinement
3.1.2. Sprint Goal Definition
3.1.3. Regular Monitoring and Adjustment
3.2. Sprint 3 :User Registration and Program Enrollment
3.2.1. Sprint Back Log & Planning
3.2.2. User RegistrationModule
3.2.3. Program Enrollment Module
3.2.4 Unit Testing
3.2.5. Integration Testing
3.2.6. User Acceptance Testing
3.2.7. Retrospective Meeting for Process Improvement
3.2.8. Deployment
3.3. Sprint 4:Purchase Plant and Payment
3.3.1. Sprint Back Log & Planning
3.3.2. Plant Purchase Module
3.3.3. Payment Processing Module
3.3.4 Unit Testing
3.3.5. Integration Testing
3.3.6. User Acceptance Testing
3.3.7. Retrospective Meeting for Process Improvement
3.3.8. Deployment
3.4. Sprint 5: Forum & Report Generation
3.4.1. Sprint Back Log & Planning
3.4.2. Forum Interface Design
3.4.3. Report Generation
3.4.4 Unit Testing
3.4.5. Integration Testing
3.3.6. User Acceptance Testing
3.3.7. Retrospective Meeting for Process Improvement
3.3.8. Deployment
3.5. Sprint 6:Testing
3.5.1. System Testing
3.5.2. User Acceptance Testing
3.5.3. Compatibility Testing
3.5.4. Usability Testing
4. Sprint 7: Finalization and Deployment
4.1 Expanded Use Case Descriptions
4.2 Collaboration Diagram
4.3 Sequence Diagram
4.4 Class Diagram
4.5 Testing and Validation
5. Ongoing Maintenance and Support
5.1 Post-Deployment Support
5.2 Continuous Improvement

Figure 1: Work Breakdown Structure

The provided Work Breakdown Structure(WBS) for McGregor PlantSphere is aligned with Agile Methodology. This can be explaned from following aspects:

1.Sprint-based Structure: The WBS is organised into sprints each representing a specific phase or iteration of development where work is broken down into managable chunks called sprints.

2.Iterative Development: The WBS includes multiple iterations or sprints for different phases such as project initiation,analysis and planning,iterative development sprints,testing,finalization,and deployment.

3.Incremental Delivery:The WBS suggests delivering the product incrementally through the completion of various sprints,starting from project initiation and progressing through analysis,development,testing,finalization and deployment,testing,finalization,and deployment.

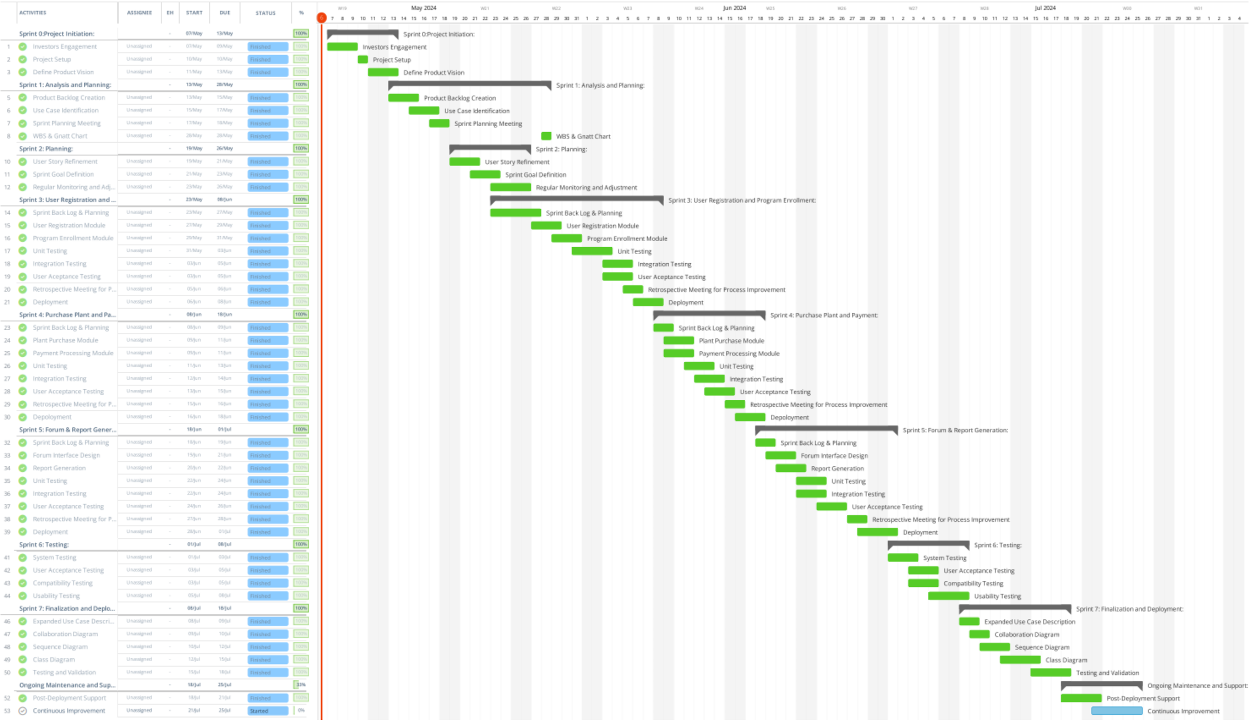
4.Emphasis on Collaboration and Feedback: The inclusion of tasks such as sprint planning meetings,retrospective meetings for process improvement,and continous improvement indicates a focus on collaboration,feedbacks and adaptation which are fundamental aspects of agile methodology.

3.Gantt Chart :

A Gantt chart, commonly used in project management, is one of the most popular and useful ways of showing activities (tasks or events) displayed against time. On the left of the chart is a list of the activities and along the top is a suitable time scale. Each activity is represented by a bar; the position and length of the bar reflects the start date, duration and end date of the activity. This allows you to see at a glance:

- What the various activities are
- When each activity begins and ends
- How long each activity is scheduled to last
- Where activities overlap with other activities, and by how much
- The start and end date of the whole project

(Gantt, 2021)



		ACTIVITIES	ASSIGNEE	EH	START	DUE	STATUS	%
		Sprint 0:Project Initiation:		-	07/May	13/May		100%
1	✓	Investors Engagement	Unassigned	-	07/May	09/May	Finished	100%
2	✓	Project Setup	Unassigned	-	10/May	10/May	Finished	100%
3	✓	Define Product Vision	Unassigned	-	11/May	13/May	Finished	100%
		Sprint 1: Analysis and Planning:		-	13/May	28/May		100%
5	✓	Product Backlog Creation	Unassigned	-	13/May	15/May	Finished	100%
6	✓	Use Case Identification	Unassigned	-	15/May	17/May	Finished	100%
7	✓	Sprint Planning Meeting	Unassigned	-	17/May	18/May	Finished	100%
8	✓	WBS & Gnatt Chart	Unassigned	-	28/May	28/May	Finished	100%
		Sprint 2: Planning:		-	19/May	26/May		100%
10	✓	User Story Refinement	Unassigned	-	19/May	21/May	Finished	100%
11	✓	Sprint Goal Definition	Unassigned	-	21/May	23/May	Finished	100%
12	✓	Regular Monitoring and Adj...	Unassigned	-	23/May	26/May	Finished	100%
		Sprint 3: User Registration and ...		-	23/May	08/Jun		100%
14	✓	Sprint Back Log & Planning	Unassigned	-	23/May	27/May	Finished	100%
15	✓	User Registration Module	Unassigned	-	27/May	29/May	Finished	100%
16	✓	Program Enrollment Module	Unassigned	-	29/May	31/May	Finished	100%
17	✓	Unit Testing	Unassigned	-	31/May	03/Jun	Finished	100%
18	✓	Integration Testing	Unassigned	-	03/Jun	05/Jun	Finished	100%
19	✓	User Aceptance Testing	Unassigned	-	03/Jun	05/Jun	Finished	100%
20	✓	Retrospective Meeting for P...	Unassigned	-	05/Jun	06/Jun	Finished	100%
21	✓	Deployment	Unassigned	-	06/Jun	08/Jun	Finished	100%
		Sprint 4: Purchase Plant and Pa...		-	08/Jun	18/Jun		100%
23	✓	Sprint Back Log & Planning	Unassigned	-	08/Jun	09/Jun	Finished	100%
24	✓	Plant Purchase Module	Unassigned	-	09/Jun	11/Jun	Finished	100%
25	✓	Payment Processing Module	Unassigned	-	09/Jun	11/Jun	Finished	100%
26	✓	Unit Testing	Unassigned	-	11/Jun	13/Jun	Finished	100%
27	✓	Integration Testing	Unassigned	-	12/Jun	14/Jun	Finished	100%
28	✓	User Acceptance Testing	Unassigned	-	13/Jun	15/Jun	Finished	100%
29	✓	Retrospective Meeting for P...	Unassigned	-	15/Jun	16/Jun	Finished	100%
30	✓	Depoloyment	Unassigned	-	16/Jun	18/Jun	Finished	100%
		Sprint 5: Forum & Report Gener...		-	18/Jun	01/Jul		100%
32	✓	Sprint Back Log & Planning	Unassigned	-	18/Jun	19/Jun	Finished	100%
33	✓	Forum Interface Design	Unassigned	-	19/Jun	21/Jun	Finished	100%
34	✓	Report Generation	Unassigned	-	20/Jun	22/Jun	Finished	100%
35	✓	Unit Testing	Unassigned	-	22/Jun	24/Jun	Finished	100%
36	✓	Integration Testing	Unassigned	-	22/Jun	24/Jun	Finished	100%
37	✓	User Acceptance Testing	Unassigned	-	24/Jun	26/Jun	Finished	100%
38	✓	Retrospective Meeting for P...	Unassigned	-	27/Jun	28/Jun	Finished	100%
39	✓	Deployment	Unassigned	-	28/Jun	01/Jul	Finished	100%
		Sprint 6: Testing:		-	01/Jul	08/Jul		100%
41	✓	System Testing	Unassigned	-	01/Jul	03/Jul	Finished	100%
42	✓	User Acceptance Testing	Unassigned	-	03/Jul	05/Jul	Finished	100%
43	✓	Compatibility Testing	Unassigned	-	03/Jul	05/Jul	Finished	100%
44	✓	Usability Testing	Unassigned	-	05/Jul	08/Jul	Finished	100%
		Sprint 7: Finalization and Deplo...		-	08/Jul	18/Jul		100%
46	✓	Expanded Use Case Descri...	Unassigned	-	08/Jul	09/Jul	Finished	100%
47	✓	Collaboration Diagram	Unassigned	-	09/Jul	10/Jul	Finished	100%
48	✓	Sequence Diagram	Unassigned	-	10/Jul	12/Jul	Finished	100%
49	✓	Class Diagram	Unassigned	-	12/Jul	15/Jul	Finished	100%
50	✓	Testing and Validation	Unassigned	-	15/Jul	18/Jul	Finished	100%
		Ongoing Maintenance and Sup...		-	18/Jul	25/Jul		33%
52	✓	Post-Deployment Support	Unassigned	-	18/Jul	21/Jul	Finished	100%
53	✓	Continuous Improvement	Unassigned	-	21/Jul	25/Jul	Started	0%

Figure 2: Gantt Chart

4.Use Case Diagram:

A use case is a description of the different ways that a user can interact with an application or product. They define the various external entities that exist outside the system, as well as the specific interactions they have with the system. This can come in the form of success scenarios, alternate paths, and more. Use cases can also be formatted in a text-based medium like a list/table, or a visual medium like a diagram.

The major benefits that use cases provide are in the planning stage of development. Steps like requirements gathering, defining scope, and roadmap creation are all improved through the clarity that these use cases bring. They also allow the team to identify the best possible outcome scenario, accurately depicting the intended design and use of the system. (Inflectra Coorporation, 2024)

Use Case Diagram – Notations

1.Use case: A use case is a single unit of meaningful work. It provides a high-level view of behavior observable to someone or something outside the system. The notation for a use case is an ellipsee

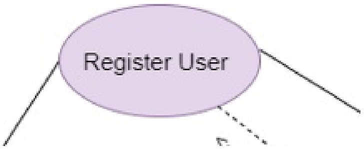


Figure 3: Use Case Notation

2.Actors: A use case diagram shows the interaction between the system and entities external to the system. These external entities are referred to as actors. Actors represent roles which may include human users, external hardware or other systems.

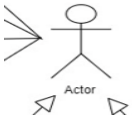


Figure 4: Actor Notation

3.<u>Communication Link</u>**:** Communication Link is used to connect actors to use cases indicating that the actor and the use case communicate with one another using messages.

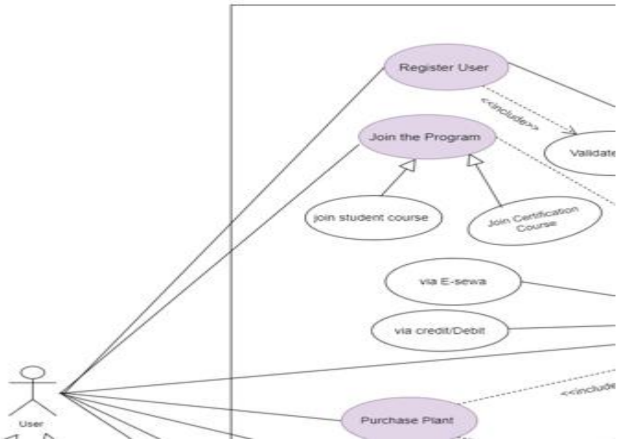


Figure 5: Communication Link

<u>4.Boundary</u>**:** It is usual to display use cases as being inside the system and actors as being outside the system.

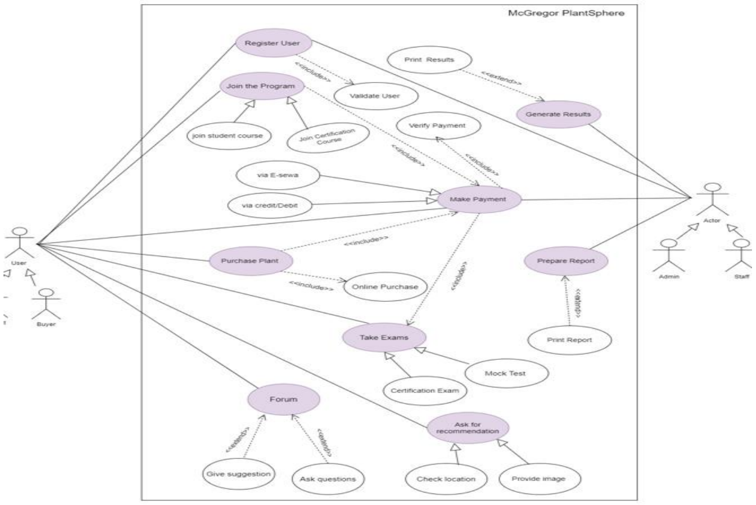


Figure 6: Boundary

5. Use Case Diagram-Relationships: There are 4 types of relationships that are:

- Association
- Generalization
- Extends
- Includes

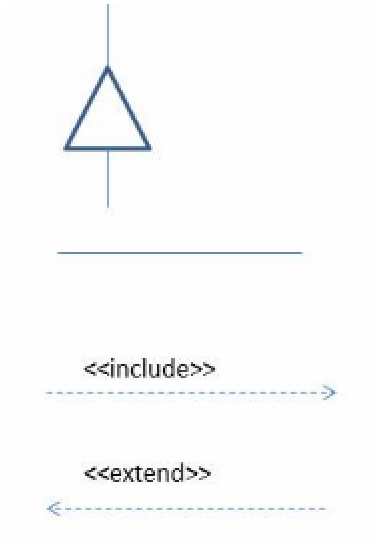


Figure 7: Use case diagram-Relationships

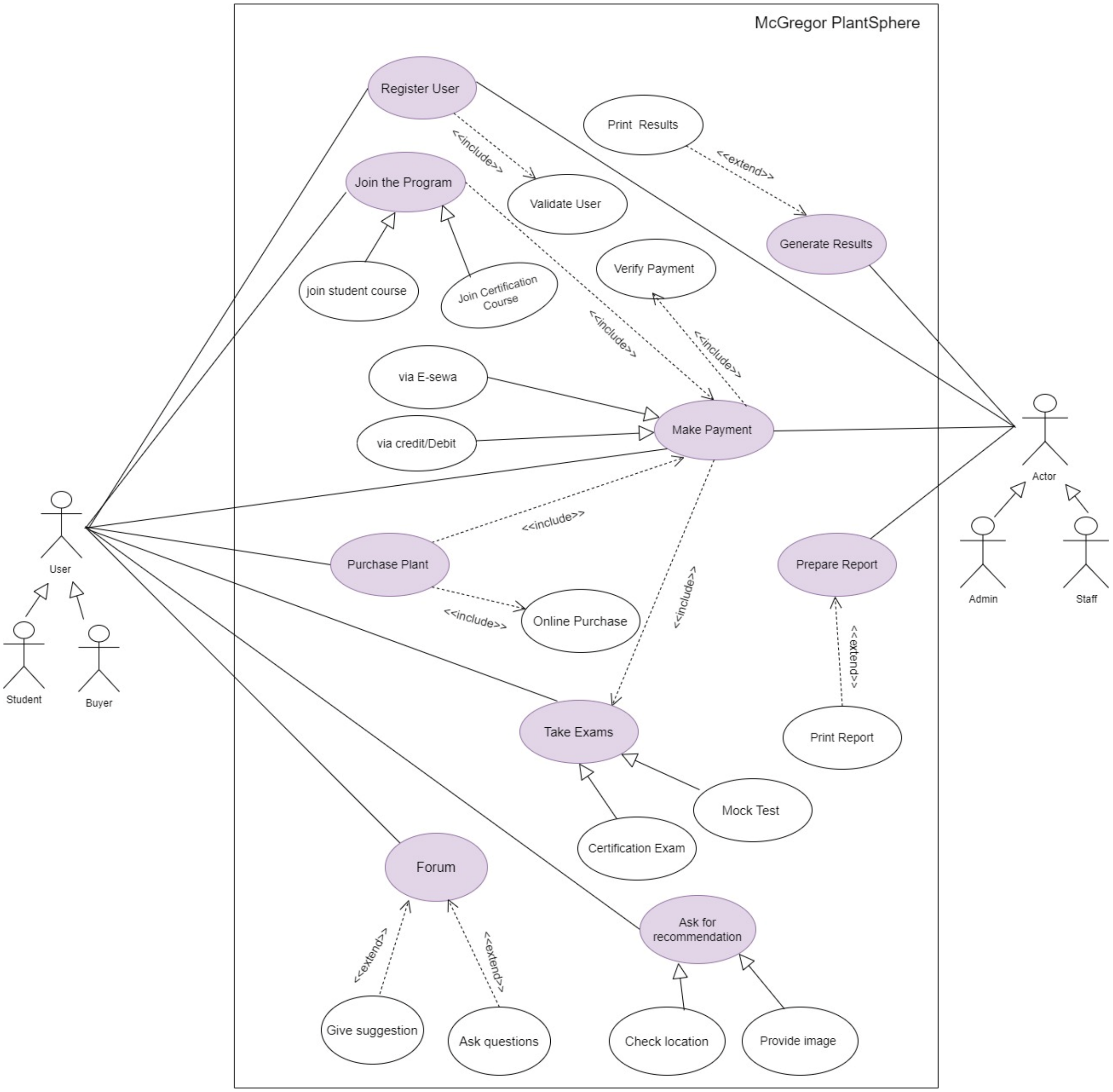


Figure 8:Use Case Diagram

5.High Level Use Case:

A high-level use case description is a paragraph or two summary of a task that is written during the late inception phase of a project. It is used to provide enough detail to understand the complexity of the task and group related use cases for development. High-level use case descriptions are useful in the early stages of software development when no detailed decisions have been made about the design of the system. These descriptions need only document the purpose of the use case, the actors involved, and give a general overview of what happens**.**

1.Use Case:Register User

Actors: New Users(Initiators)

Description : A new user provides their personal details and is validated by the system and recieves registration confirmation.

2.Use Case:Join the Program

Actors: Registered Users

Description : Any registered users who are willing to join the programs can apply for the courses through the system. After choosing the suitable courses the users are then redirected to payment.

3.Use Case: Purchase Plant

Actors: Registered Users

Description: Any registered users can purchase a plant. Users can choose any plant that is being displayed by the system which is then verified by the system for its availability and the users are requested to make payment.

4.Use Case: Make Payment

Actors: Registered Users

Description: Registered Users are redirected to payment after choosing a suitable course or purchasing a plant.Payment is then verified by the system and receipt is sent to the user after successful payment.

5. Use Case: Take Exams

Actors: Users

Description : Users can take either mock exam or certification exam according to their choice only after payment.

6. Use Case:Forum

Actors: Users

Description: User can discuss about their issues, share their experience and ideas as well as connect with people all around the world. Users can ask questions as well as give suggestions.

7.Use Case:Ask for Recommendation

Actors : Users

Description : User can provide images of the soil or provide their location for the expertise to help them out.

8.Use Case: Prepare reports

Actor : Admins

Description: Admin prepare reports regarding the detailed financial report, employees report and report related to every user.

9.Use Case: Generate Results

Actor: Admin

Description: Admin generate results for the exams that are given by the user.The result include the marks they obtained with their details.

6.Expanded Use Case:

1.Register User

Use Case: Register User

Actor: User

Description : User registers after clicking the register button in home page which then redirects user to the registration page.The user then fills the form with necessary details and submits. The system then verifies the user and successfull registration message is redirected to the user.

Actor Action	System Response
1.User clicks the register button	
	2. Redirects user to registration form.
3.Fills the registration form with all the necessary personal details.	
4.Submits the registration form.	
	5.Validates the user.
	6.Displays successful registration message
7.Receives successful registration message.	

Table 1:Expandad Use Case – Registration

Alternate Cases:

Line 5: If the user input is not valid the system displays error message and ends use case.

2.Purchase Plants:

Use Case: Purchase Plants

Actor: Customer

Description: The customer purchases the plant from shop section after the user is registered. The user then selects the plant they want to buy and and click the buy button. The system then checks the availability of plant and display. If the plant is available it request payment . User then makes payment and the payment is verified by the system and stores the payment details and displays successful message.

Actor Action	System Response
	1.System displays the shop section after registration.
2.User explores the shop and selects the item.	
3.Clicks on the purchase button.	
	4.Checks the availability of plant and request payment.
5.Makes payment for the plant.	
	6.Verifies the payment and stores payment details
	7.Displays successful purchase message
8.Receives successful purchase message.	

Table 2: Expanded Use Case- Purchase Plant

Line 4: System checks for the availability of plant and if it is unavailable System displays failure and ends use case.

Line 6: System verifies the payment and if it is not verified System displays failure and ends use case.

7.Communication Diagram:

A communication diagram is an extension of object diagram that shows the objects along with the messages that travel from one to another. In addition to the associations among objects, communication diagram shows the messages the objects send each other.

Purpose of Communication Diagram

- Model message passing between objects or roles that deliver the functionalities of use cases and operations
- Model mechanisms within the architectural design of the system
- Capture interactions that show the passed messages between objects and roles within the collaboration scenario
- Model alternative scenarios within use cases or operations that involve the collaboration of different objects and interactions
- Support the identification of objects (hence classes), and their attributes (parameters of message) and operations (messages) that participate in use cases

7.1.Sequence Diagram:

One of the primary uses of sequence diagrams is in the transition from requirements expressed as use cases to the next and more formal level of refinement. Use cases are often refined into one or more sequence diagrams. In addition to their use in designing new systems, sequence diagrams can be used to document how objects in an existing system currently interact

The main purpose of a sequence diagram is to define event sequences that result in some desired outcome. The focus is less on messages themselves and more on the order in which messages occur; nevertheless, most sequence diagrams will communicate what messages are sent between a system's objects as well as the order in which they occur. The diagram conveys this information along the horizontal and vertical dimensions: the vertical dimension shows, top down, the time sequence of messages/calls as they occur, and the horizontal dimension shows, left to right, the object instances that the messages are sent to.

Sequence Diagram-Notations:

- Lifeline Notation: A sequence diagram is made up of several of these lifeline notations that should be arranged horizontally across the top of the diagram. No two lifeline notations should overlap each other. They represent the different objects or parts that interact with each other in the system during the sequence. A lifeline notation with an actor element symbol is used when the particular sequence diagram is owned by a use case.

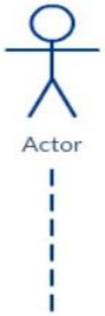


Figure 9: Lifeline Notation

- Activation Bars : The activation bar is the box placed on the lifeline. It is used to indicate that an object is active (or instantiated) during an interaction between two objects. The length of the rectangle indicates the duration of the objects staying active.

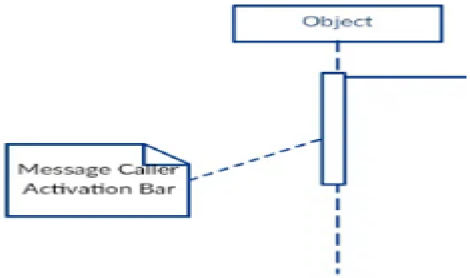
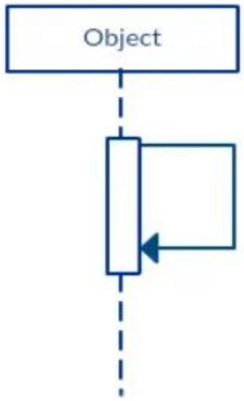


Figure 10: Activation Bar

- Message Arrows :An arrow from the Message Caller to the Message Receiver specifies a message in a sequence diagram. A message can flow in any direction; from left to right, right to left, or back to the Message Caller itself.

While you can describe the message being sent from one object to the other on the arrow, with different arrowheads you can indicate the type of message being sent or received.There are different types of messages.They are: Synchronous message, Asynchronous message,Return Message,Creation Message,Reflexive Message (IBM Developer, n.d.)



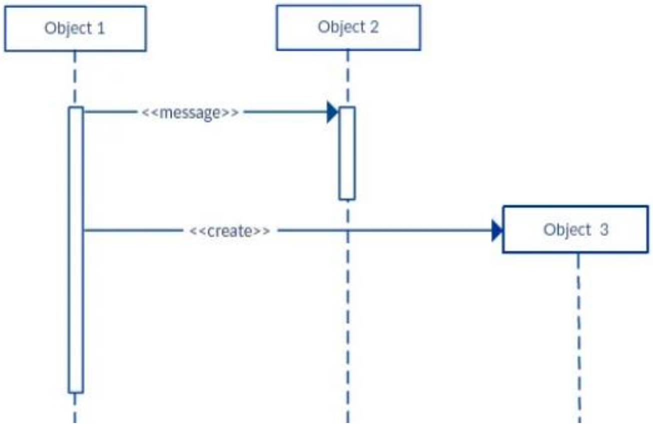


Figure 11: Creation Message

Figure 12: Reflexive Message

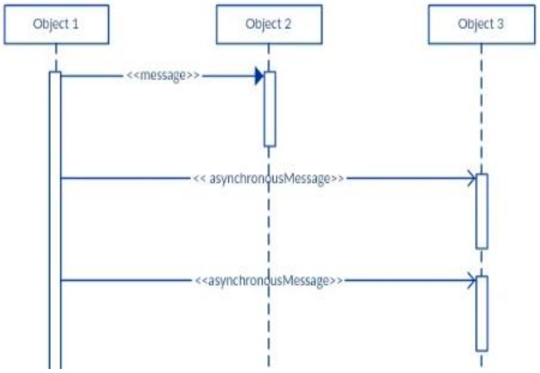


Figure 13:Synchronous & Asynchronous Message

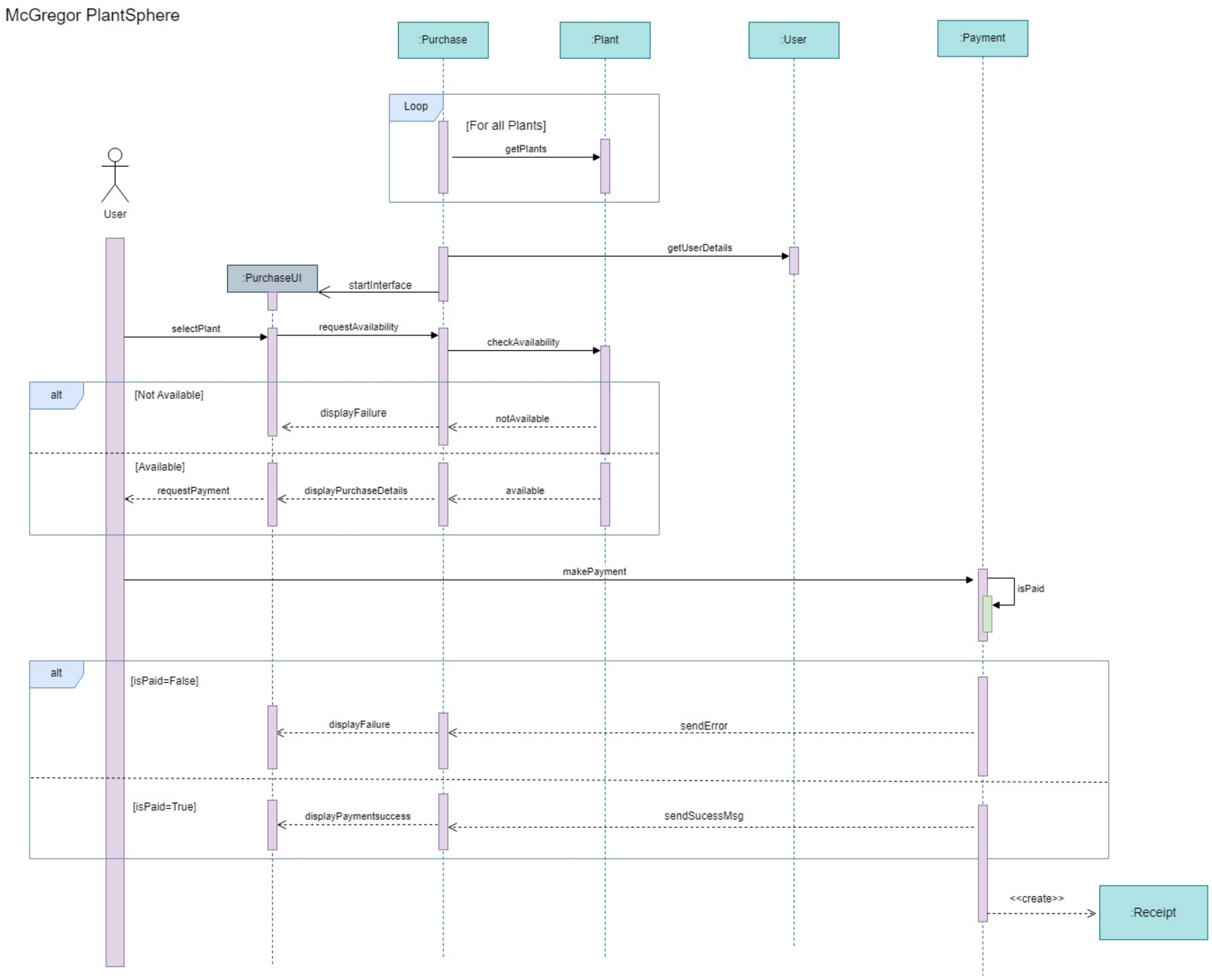


Figure 14: Sequence Diagram

Description: In the scenario, the sequence diagram started through drawing the user as the actor and setting up its lifeline for the Purchase feature. The controller, called 'Purchase', had 'PurchaseUI' as its interface. We listed four objects followed by a semicolon, excluding the one that was created later. We then showed the flow and activation bars in a straightforward way, following the system's sequence, with clear and propoer messages.

7.2.Collaboration Diagram:

Collaboration diagrams are used to show how objects interact to perform the behavior of a particular use case, or a part of a use case. Along with sequence diagrams, collaboration are used by designers to define and clarify the roles of the objects that perform a particular flow of events of a use case. They are the primary source of information used to determining class responsibilities and interfaces.

Unlike a sequence diagram, a collaboration diagram shows the relationships among the objects. Sequence diagrams and collaboration diagrams express similar information, but show it in different ways.

Benefits of a Collaboration Diagram

1. The collaboration diagram is also known as Communication Diagram.
2. It mainly puts emphasis on the structural aspect of an interaction diagram, i.e., how lifelines are connected.
3. The syntax of a collaboration diagram is similar to the sequence diagram; just the difference is that the lifeline does not consist of tails.
4. The messages transmitted over sequencing is represented by numbering each individual message.
5. The collaboration diagram is semantically weak in comparison to the sequence diagram.

Collaboration Diagram-Notations:

- Objects: The representation of an object is done by an object symbol with its name and class underlined, separated by a colon.

- Actors**:** In the collaboration diagram, the actor plays the main role as it invokes the interaction. Each actor has its respective role and name. In this, one actor initiates the use case.

- Links**:** The link is an instance of association, which associates the objects and actors. It portrays a relationship between the objects through which the messages are sent. It is represented by a solid line. The link helps an object to connect with or navigate to another object, such that the message flows are attached to links.

- Messages**:** It is a communication between objects which carries information and includes a sequence number, so that the activity may take place. It is represented by a labeled arrow, which is placed near a link. The messages are sent from the sender to the receiver, and the direction must be navigable in that particular direction. The receiver must understand the message.

(Visual-Paradigm, 2022)

Description: In this scenario, we created a collaboration diagram for the Purchase feature. The main class we focused on is 'Purchase'. We identified four objects within this feature, excluding one that gets created based on certain conditions. The controller object is labeled as 'Purchase', overseeing the process. The actor, representing the user, is also included. Lines showed the relationships between objects, while arrows depicted the flow of methods.

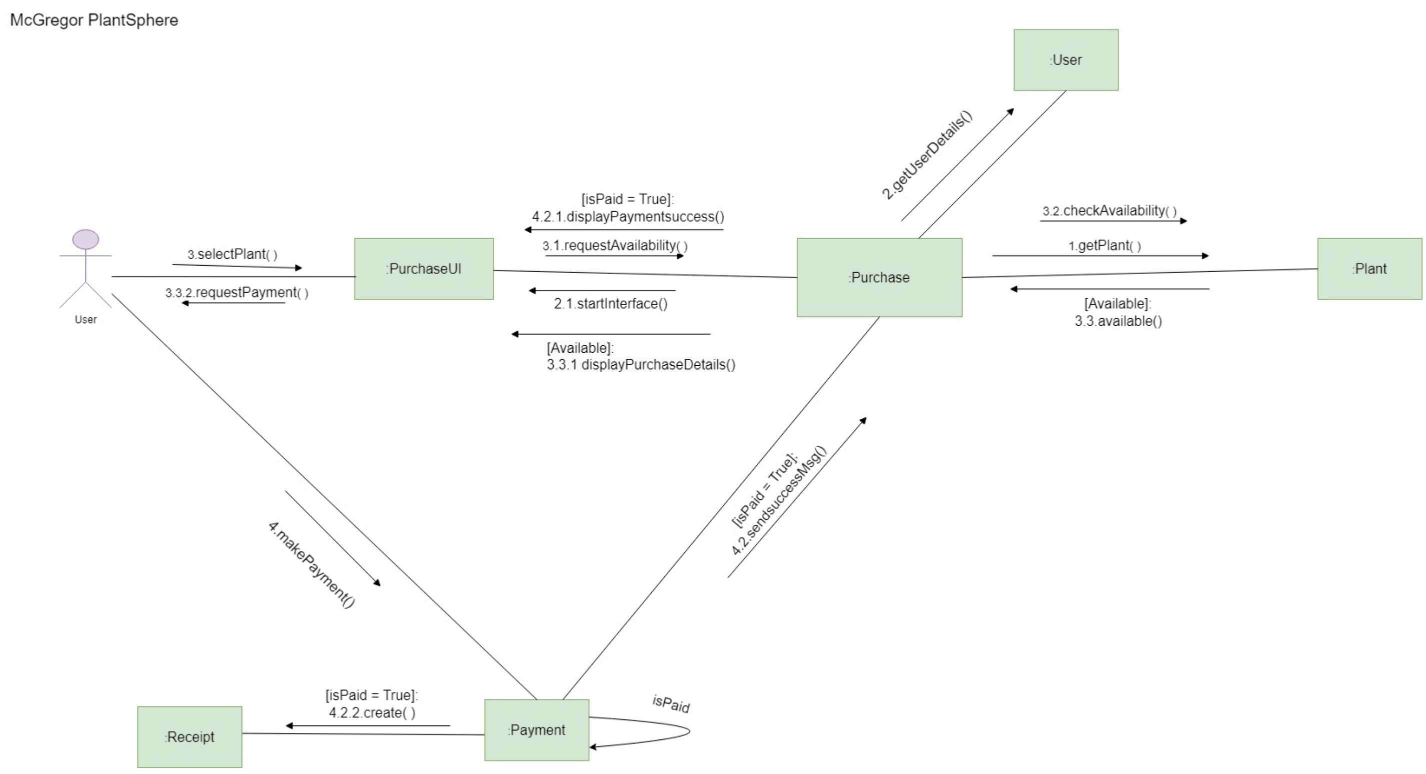


Figure 15: Collaboration Diagram

8.Class Diagram:

A class diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

Purpose of Class Diagrams

1. Shows static structure of classifiers in a system
2. Diagram provides a basic notation for other structure diagrams prescribed by UML
3. Helpful for developers and other team members too
4. Business Analysts can use class diagrams to model systems from a business perspective

Class Diagram-Notations:

- Class Name: The name of the class appears in the first partition
- Class Attributes : The attributes are shown in the second partition.It is shown after a colon
- Class Operations: Operations are shown in the third partition
- Class Relationships: A class may be involved in one or more relationships with other classes. A relationship can be one of the following types

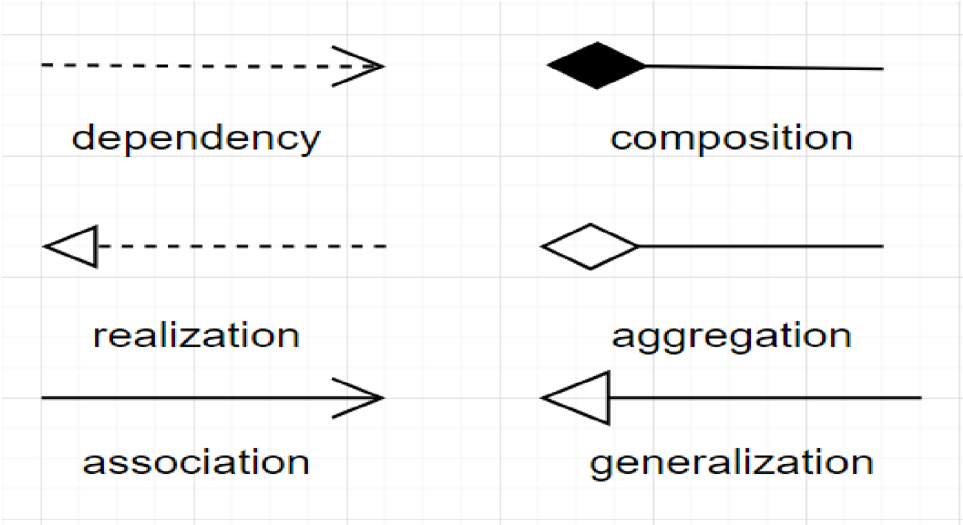


Figure 16: Class Diagram - Notations

(Visual-Paradigm, 2022)

During the development of class diagram the uniquely identified classes are as follows:

1.User
2.Admin
3.Staff
4.Registration
5.Login
6.Purchase
7.Payment
8.Exam
9.Result
10.Report

Use cases and their domain classes:

Use Cases	Domain Classes
Register User	Registration, User
Join the program	User, Courses
Purchase Plant	User, Purchase
Make Payment	User, Payment
Take exams	User, Exam
Generate results	Staff, Results
Prepare report	Admin, Report
Forum	User, Admin
Ask for Recommendation	User, Admin

Table 3: Use case and their Domain Classes

Description: In the Class Diagram below, classes are positioned at the top, displaying their attributes denoted by the (-) sign, followed by methods indicated by the (+) sign. The relationships between classes are depicted: a one-to-one relationship exists between 'registration' and 'user', meaning each user is associated with one registration form. Conversely, a zero-to-many relationship is shown between 'user' and 'courses', 'exams', and 'purchase', indicating that a user can interact with multiple courses, exams, and purchases, but these interactions are not mandatory. Similarly, one user can view details for one result, and one payment transaction generates one receipt. Staff members manage the store, with one admin overseeing multiple staff members. This simplified representation highlights the connections and interactions between classes in the system.

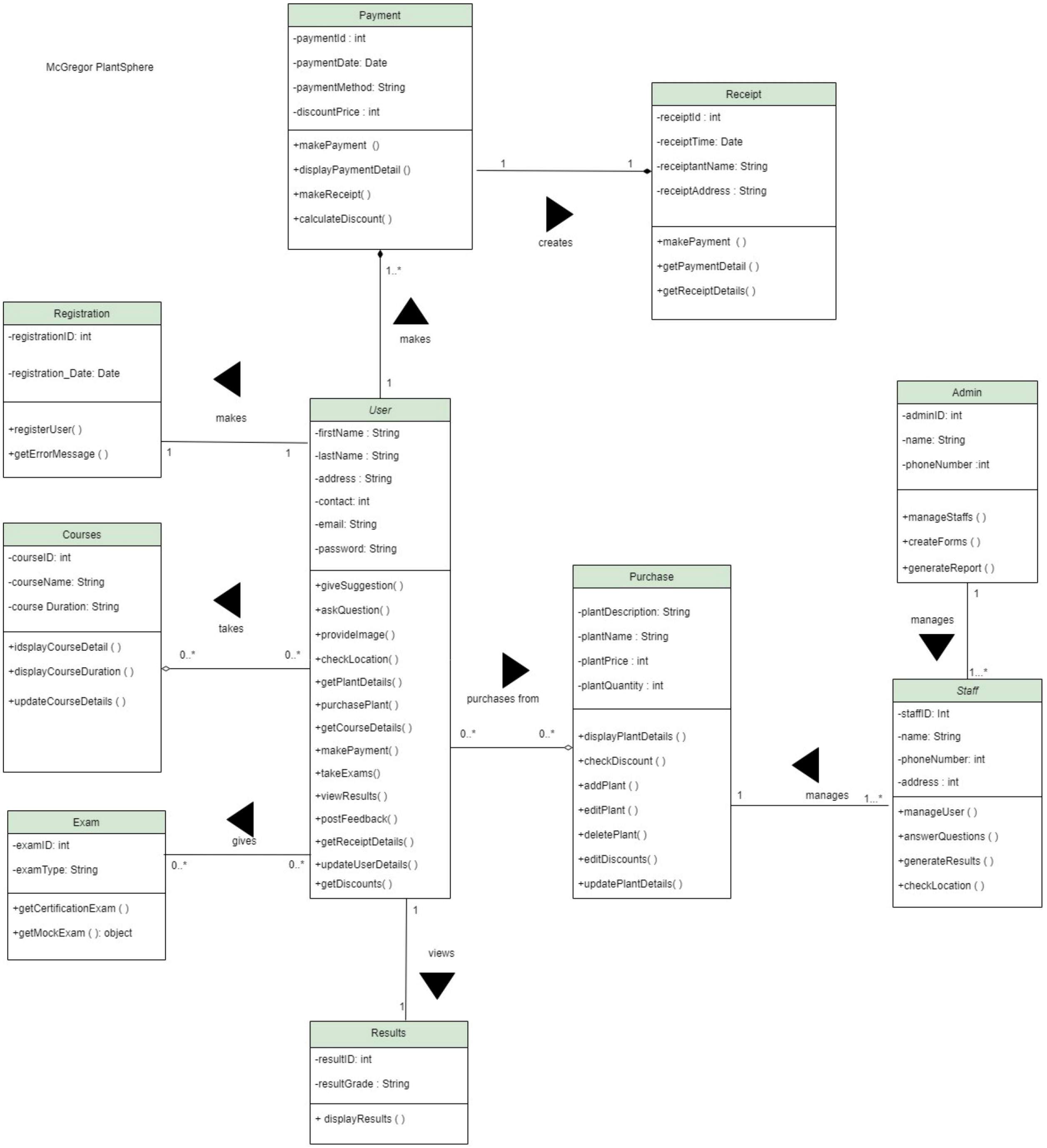


Figure 17: Class Diagram

9.Further Development:

9.1.Methodology Used:

The methodology used in the development of McGregor PlantSphere is Agile Methodology.The choice of this methodology for its development is driven by its flexibility,adaptability and focus on delivering value to stakeholders in a dynamic and rapidly changing environment. Agile's iterative and incremental approach allows for the frequent delivery of working software, enabling stakeholders to provide feedback early and often throughout the development process. By breaking down the project into manageable sprints, Agile facilitates continuous collaboration between the development team and stakeholders, ensuring that the evolving needs and requirements of McGregor PlantSphere are effectively addressed. Additionally, Agile methodology promotes a culture of transparency, communication, and shared responsibility, empowering the team to make data-driven decisions and respond quickly to changes in priorities or market conditions.

Overall, Agile methodology aligns closely with McGregor PlantSphere's goal of building a robust and user-centric platform for botanical enthusiasts, enabling the project to adapt and evolve iteratively in response to stakeholder feedback and market dynamics.

9.2.Architectural Choice:

Microservices Architecture

The selection of microservices architecture for McGregor PlantSphere is driven by several key considerations that align with the system's requirements and objectives.The adoption of microservices architecture for McGregor PlantSphere is driven by its scalability, flexibility, resilience, and support for continuous delivery practices.

McGregor PlantSphere aims to be a scalable and extensible platform that can accommodate a growing user base and evolving business requirements. Microservices architecture facilitates scalability by breaking down the system into smaller,independently deployable services. Each service can be developed, deployed, and scaled independently, allowing McGregor PlantSphere to scale horizontally by adding more instances of individual services as needed.It is expected to integrate with external services, such as payment gateways, plant databases, and shipping providers. Each microservice can encapsulate a specific functionality or business logic, enabling easy integration with external systems without impacting the overall system architecture. This flexibility in integration ensures that McGregor PlantSphere can leverage external services to enhance its functionality and provide a richer user experience.Microservices architecture supports continuous integration and continuous deployment , where changes to individual services can be tested, validated, and deployed independently.

By leveraging microservices, McGregor PlantSphere can build a modular, scalable, and resilient platform that meets the evolving needs of its users and stakeholders while enabling rapid innovation and growth.

9.3.Design Pattern:

Model-View-Controller (MVC) pattern.

Among the design patterns mentioned, the most suitable option for McGregor PlantSphere is the Model-View-Controller (MVC) pattern.

McGregor PlantSphere, being a web-based platform, requires a robust and maintainable architecture to handle its diverse functionalities such as user registration, program enrollment, plant purchase, forum integration, and report generation. The MVC pattern provides a well-established and widely adopted architectural framework that separates the application into three interconnected components: Model, View, and Controller.

Model: In McGregor PlantSphere, this could include data structures for user information, plant details, forum posts, and reports. The Model encapsulates the application's data and defines the operations that can be performed on that data, ensuring data integrity and consistency across the system.

View: In McGregor PlantSphere, Views would correspond to the various web pages and user interfaces through which users interact with the system.

Controller:In McGregor PlantSphere, Controllers would process user requests, interact with the Model to retrieve or manipulate data, and render appropriate Views in response. The Controller helps maintain separation of concerns and promotes modular and reusable code.

The MVC pattern facilitates scalability by enabling independent development and deployment of each component. As McGregor PlantSphere grows and evolves, developers can scale individual components (Model, View, Controller) independently to accommodate increasing user demands and system requirements. It allows for flexibility in user interface design and presentation. McGregor PlantSphere may require different views or interfaces for various user interactions (e.g., user registration, plant purchase, forum browsing). The separation of concerns provided by it enables developers to modify or extend the user interface without affecting the underlying data or business logic. By separating concerns and promoting a clear division of responsibilities, it enhances code maintainability i.e. changes to one component can be implemented without affecting other components

9.4.Development Plans:

The development plan for McGregor PlantSphere is meticulously designed to ensure efficient progress and timely delivery of the project while adhering to Agile principles and methodologies. The plan is structured around iterative development sprints, each focusing on specific functionalities and objectives to incrementally build and enhance the platform.

Tools/Resources/Programming Platforms:

1. Version Control & Collaboration:

- GitHub for code hosting and collaboration.

2. Project Management & Agile Development:

- Agile methodologies for iterative development.
- Sprint for organizing development iterations and daily stand-up meetings.

3. Frontend Development:

- JavaScript, HTML, CSS for frontend development.
- CodePen for building interactive user interfaces.

4. Backend Development: Java and Vs Code for backend development.
5. Database: MySQL localhost for database management.
6. Testing & Quality Assurance :

- Unit Testing: Jest or Mocha for unit testing of individual components
- Integration Testing: Selenium or Cypress for end-to-end testing of integrated components.
- User Acceptance Testing (UAT): Manual testing by users to validate system functionality.
- Compatibility Testing: Testing on different browsers and devices to ensure cross-compatibility
- Usability Testing: User feedback sessions to assess ease of use and user experience

.

Features of the System:

1. User Registration & Authentication: Implemented user registration and authentication functionalities to allow users to create accounts and login securely.

2. Program Enrollment Module: Developed a module for users to enroll in courses offered by McGregor PlantSphere.

3. Purchase Plant & Payment Processing: Implemented a feature for users to purchase plants from the platform and process payments securely.

4. Forum Integration: Integrated a forum functionality to allow users to discuss topics related to agriculture, horticulture, and plant care.

5. Report Generation: Developed a feature for generating reports, including financial reports, employee reports, and user activity reports.

6. System Testing & Quality Assurance: Developed a feature for generating reports, including financial reports, employee reports, and user activity reports.

7. Deployment & Post-Deployment Support: Deployed McGregor PlantSphere to a production environment and provide post-deployment support to address any issues or bugs identified after launch.

9.5. Test Plans:

The tests for McGregor PlantSphere includes a mix of both white box and back box testing :

1. **WhiteBox Testing** : It is a testing technique where the tester has access to the internal code structure, design, and implementation details of the software being tested.white box testing is to validate the correctness of the internal code logic and ensure that all code paths are exercised, leading to maximum coverage of the software's functionality.

- Unit Testing:
  <u>Objective</u>: Verify the functionality of individual units or components.
  <u>Description</u>: Unit tests can ensure that functions responsible for user registration, payment processing, or report generation perform as expected without relying on external dependencies. It helps identify and fix bugs early in the development process, ensuring that each component of McGregor PlantSphere functions correctly and produces the expected output.

2. **BlackBox Testing**: Black box testing, also known as behavioral testing or functional testing, is a testing technique where the tester has no access to the internal code structure, design, or implementation details of the software being tested. The primary objective of black box testing is to validate the software's functionality, usability, and conformance to requirements from an end user's perspective.

- User Acceptance Testing(UAT) :
  <u>Objective</u>: Validates the system from the perspective of end users.
  <u>Description</u>: It involves inviting users to interact with McGregor PlantSphere and provide feedback on its usability, functionality, and overall user experience. By involving users in the testing process, the development team can gather valuable feedback to identify usability issues, improve user flows, and enhance the overall user experience of the platform.

3. **Both BlackBox and WhiteBox**:

- Integration Testing :
  Objective : Validate the interactions and integration between different modules or components of the system.
  Description: It can ensure that the user registration module communicates correctly with the database or that the payment processing module interacts seamlessly with the external payment gateway.It ensures that McGregor PlantSphere functions as a cohesive system, with different features and functionalities interacting smoothly with each other.

- System Testing:
  Objective: Conduct comprehensive testing of McGregor Plantsphere system as a whole
  Description: System tests validate the functionality, performance, and reliability of McGregor PlantSphere across all its components and modules. This includes testing features such as user registration, program enrollment, plant purchase, forum integration, and report generation to ensure they meet user requirements and perform as expected.

9.6. Maintainance Plans:

Maintenance plans outline proactive strategies and activities aimed at identifying, addressing, and preventing issues that may arise during the operation of the system.The purpose of maintenance plans is to ensure the continued functionality, reliability, performance, and security of a software system or application over its entire lifecycle.

1. Post-Deployment Support (mentioned in WBS: 5.1)

Objective: Provide ongoing support and assistance to users after the system has been deployed.

Activities:

Establish a dedicated support team to address user inquiries, issues, and feedback promptly.

Implement a ticketing system or helpdesk platform to manage and prioritize support requests effectively.

Regularly monitor system performance and stability to identify and resolve any issues or disruptions.

Tools/Resources: Helpdesk software, monitoring tools, communication channels (e.g., email, chat).

2. Continuous Improvement (mentioned in WBS: 5.2)

Objective: Continuously enhance and optimize McGregor PlantSphere to meet evolving user needs and technological advancements.

Activities:

Gather feedback from users through surveys, feedback forms, and user analytics to identify areas for improvement.

Prioritize and implement updates, enhancements, and new features based on user feedback and strategic goals.

Conduct regular code reviews, performance evaluations, and security assessments to maintain system integrity and reliability.

Tools/Resources: Feedback collection tools, version control system (e.g., Git), issue tracking software, continuous integration/continuous deployment pipelines.

3. Monitoring and Adjustment (mentioned in WBS: 3.1.3)

Objective: Monitor system performance, usage patterns, and user feedback to proactively identify and address potential issues or areas for improvement.

Activities:

Set up monitoring tools and alerts to track key performance metrics, such as response times, error rates, and system availability.

Analyze user behavior and engagement data to identify trends, usage patterns, and potential usability issues.

Regularly review system logs, error reports, and support tickets to diagnose and troubleshoot any issues that arise.

Tools/Resources: Monitoring tools (e.g. Datadog), analytics platforms (e.g., Google Analytics),automated alerting systems.

4. Documentation and Knowledge Sharing

Objective: Maintain comprehensive documentation and knowledge resources to facilitate system understanding, troubleshooting, and onboarding.

Activities:

Create and update user manuals, technical documentation, and training materials to guide users and administrators.

Establish a knowledge base or docs to document common issues, best practices, and troubleshooting guides.

Conduct regular training sessions or workshops to educate users and staff on new features, updates, and system usage.

Tools/Resources: Documentation platforms (e.g.,MsWord), training materials, communication tools for announcements and notifications.

5. Security Updates and Compliance

Objective: Ensure the security and compliance of McGregor PlantSphere by applying regular updates, patches, and security best practices.

Activities:

Stay informed about emerging security threats, vulnerabilities, and regulatory requirements relevant to the system.

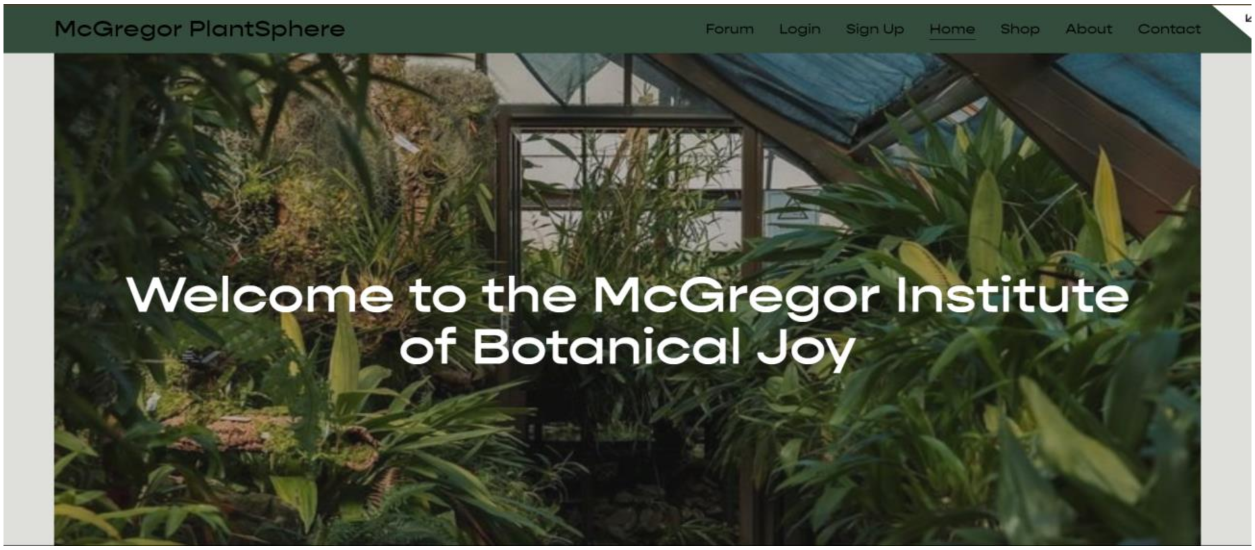
Implement timely security patches, updates, and fixes to address known vulnerabilities and mitigate potential risks.

Conduct regular security tests, and compliance assessments to validate the effectiveness of security measures.

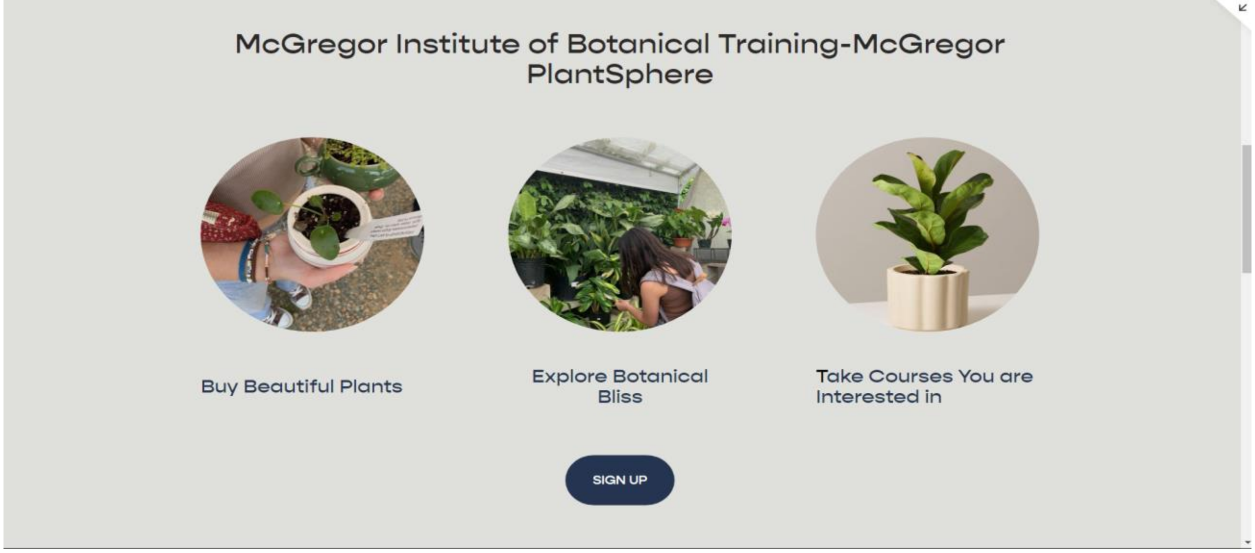
Tools/Resources: security incident response protocols.

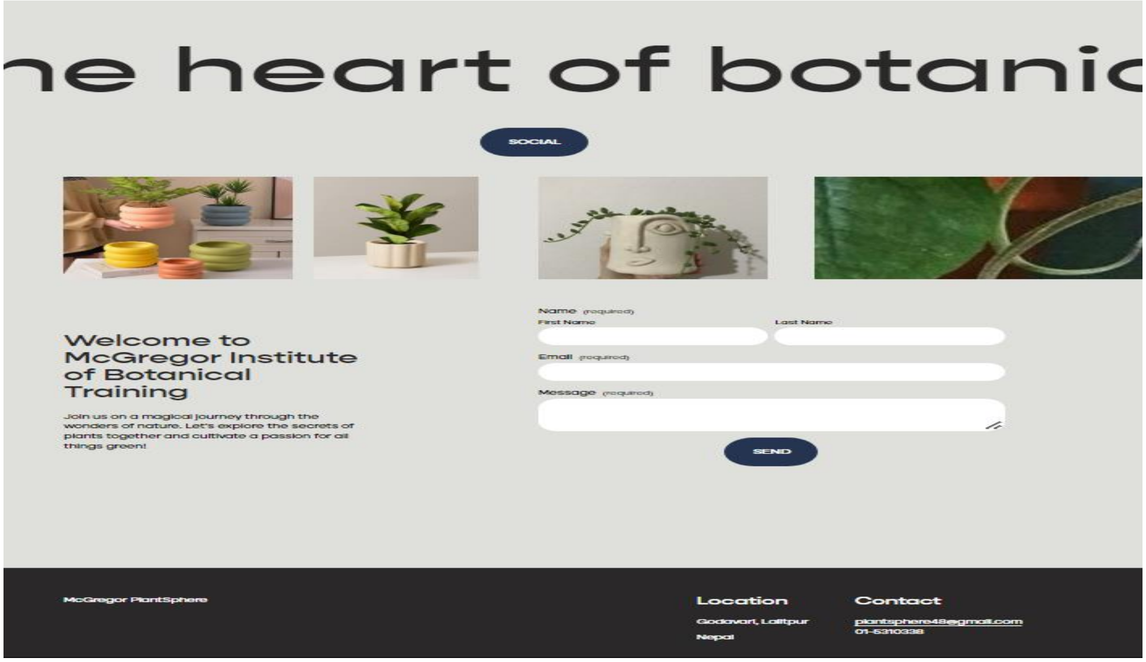
10.Prototype Development:

10.1 Home Page:



Welcome to the McGregor Institute of Botanical Joy


McGregor Institute of Botanical Training-McGregor PlantSphere



Welcome to McGregor Institute of Botanical Training

Join us on a magical journey through the wonders of nature. Let's explore the secrets of plants together and cultivate a passion for all things green!

McGregor PlantSphere

Location

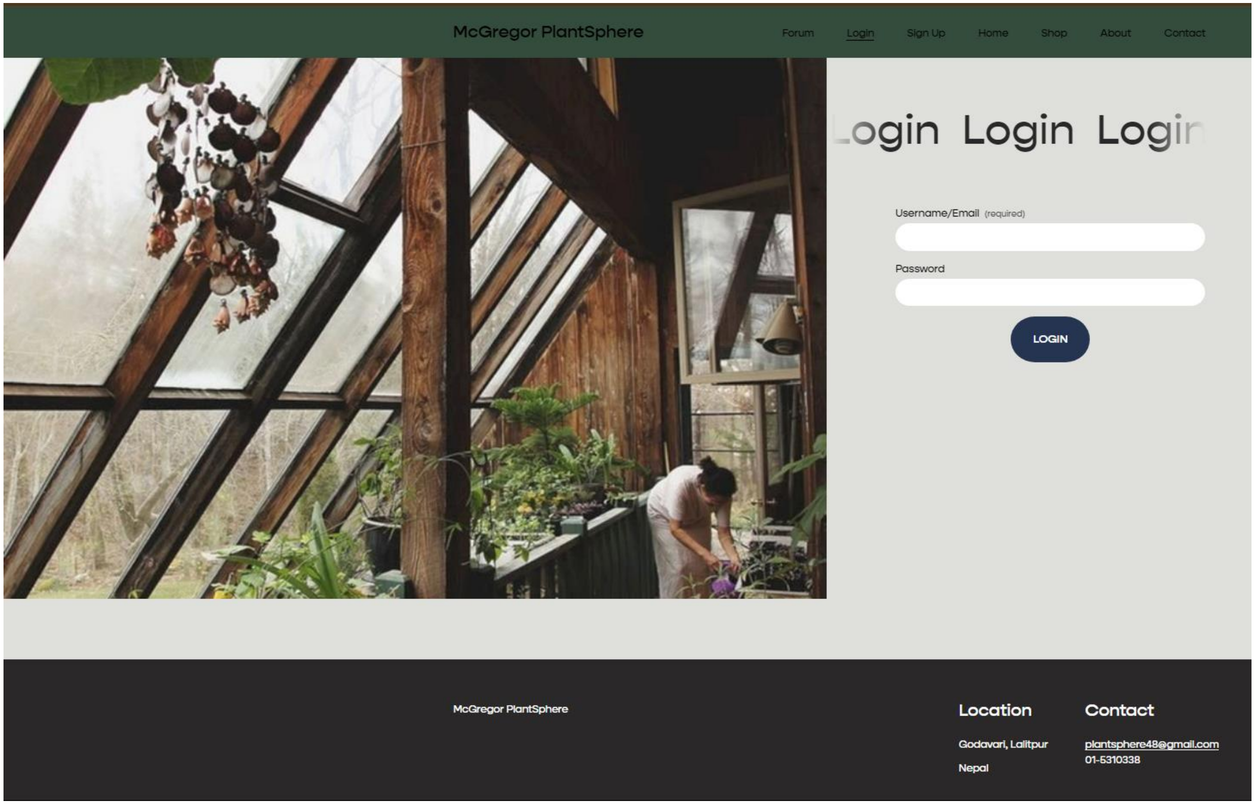
Godavari, Lalitpur
Nepal

Contact

plantsphere48@gmail.com
01-5310338

Figure 18: Home Page

10.2. Login Page:



McGregor PlantSphere

Location

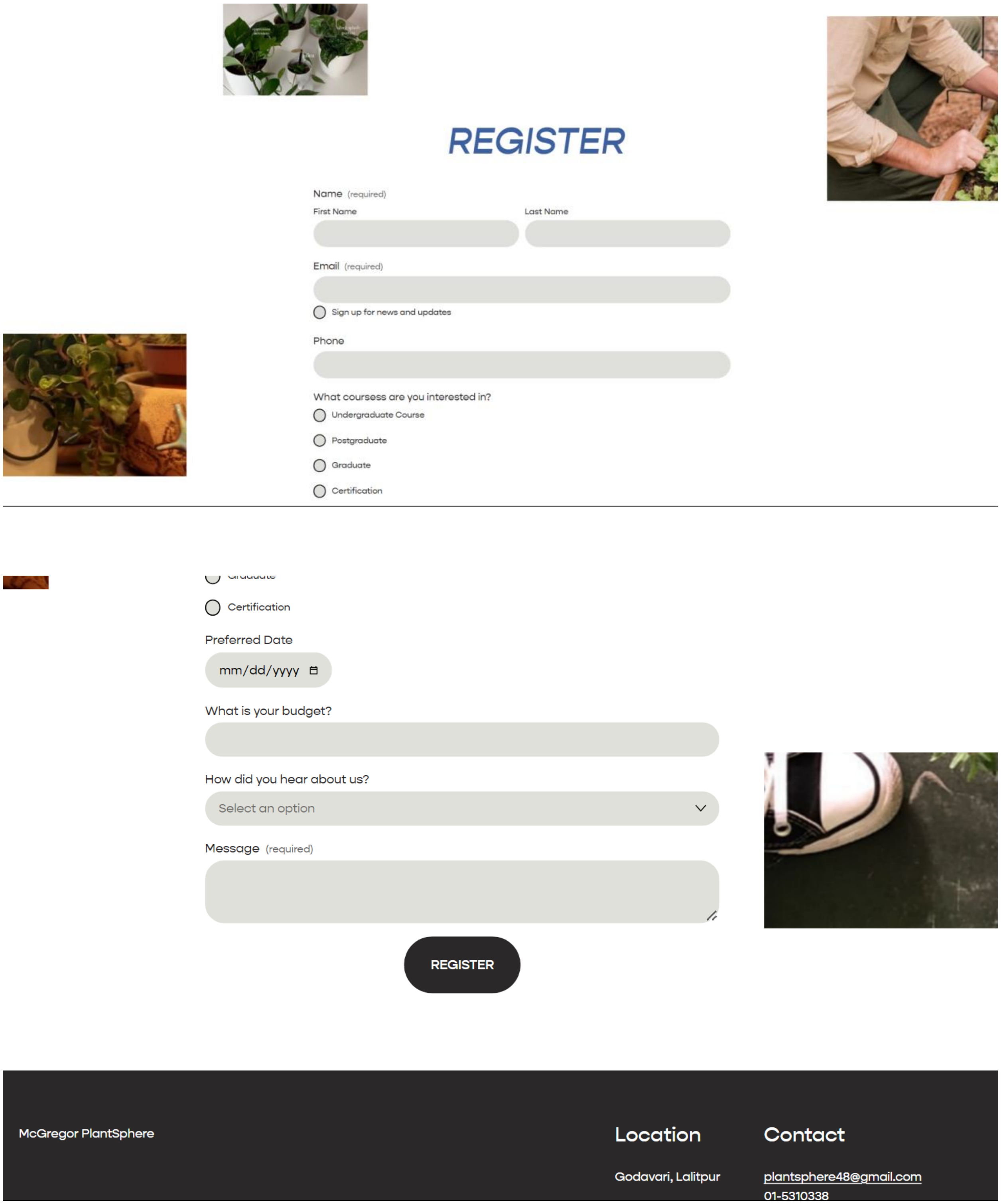
Godavari, Lalitpur
Nepal

Contact

plantsphere48@gmail.com
01-5310338

Figure 19: Login Page

10.3. Register Page



REGISTER

Name (required)

First Name

Last Name

Email (required)

○ Sign up for news and updates

Phone

What coursess are you interested in?

○ Undergraduate Course

○ Postgraduate

○ Graduate

○ Certification

Preferred Date

mm/dd/yyyy

What is your budget?

How did you hear about us?

Select an option

Message (required)

McGregor PlantSphere

Location

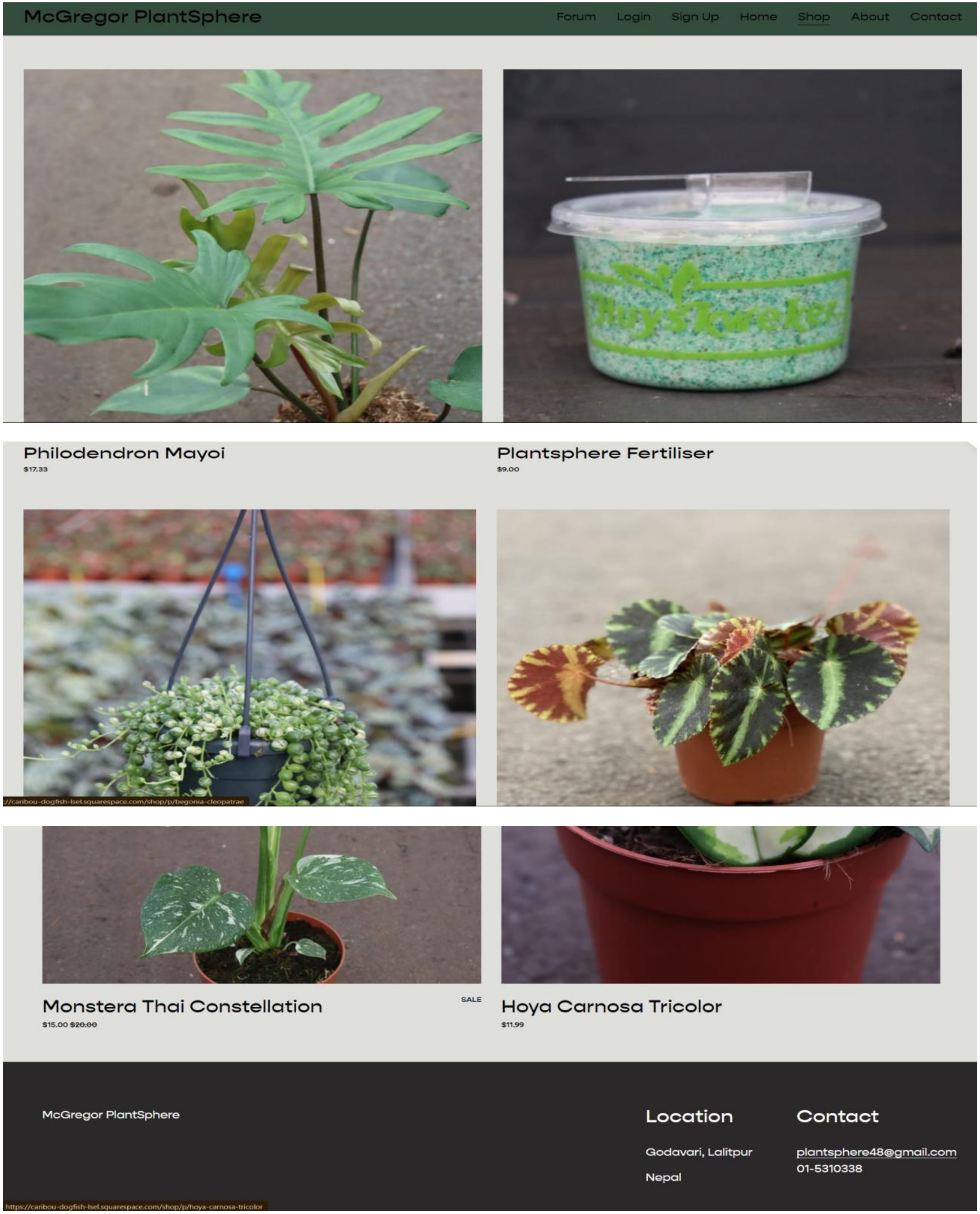
Godavari, Lalitpur

Contact

plantsphere48@gmail.com
01-5310338

Figure 20: Register Page

10.4.Product Page:


McGregor PlantSphere
Forum
Login
Sign Up
Home
Shop
About
Contact
SALE
://caribou-dogfish-lsel.squarespace.com/shop/p/begonia-cleopatrae
https://caribou-dogfish-lsel.squarespace.com/shop/p/hoya-carnosa-tricolor

Philodendron Mayoi
$17.33

Plantsphere Fertiliser
$9.00

Monstera Thai Constellation
$15.00 ~~$20.00~~

Hoya Carnosa Tricolor
$11.99

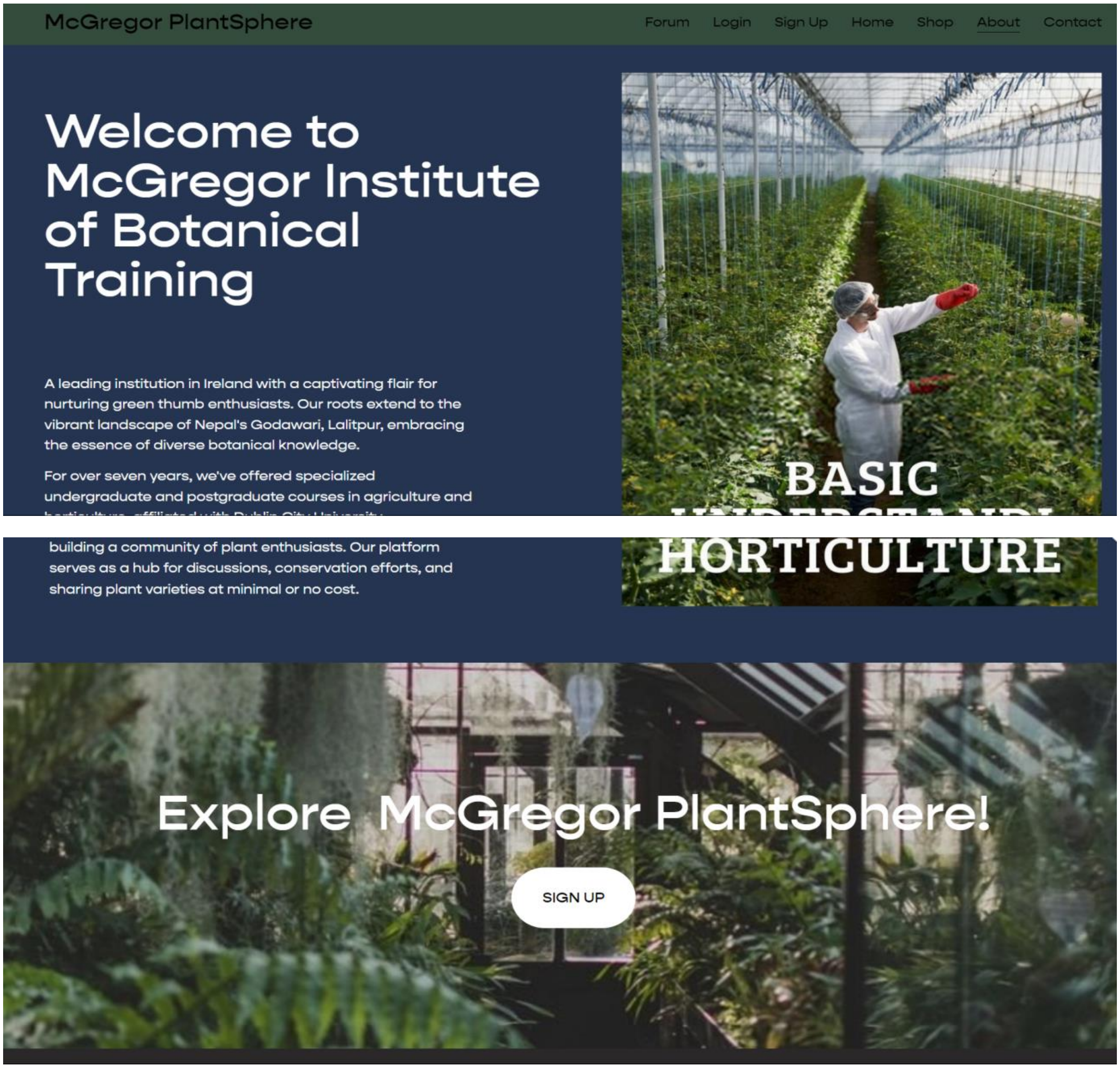
McGregor PlantSphere

Location
Godavari, Lalitpur
Nepal

Contact
plantsphere48@gmail.com
01-5310338

Figure 21: Product Page

10.5.About us :



McGregor PlantSphere

Welcome to McGregor Institute of Botanical Training

A leading institution in Ireland with a captivating flair for nurturing green thumb enthusiasts. Our roots extend to the vibrant landscape of Nepal's Godawari, Lalitpur, embracing the essence of diverse botanical knowledge.

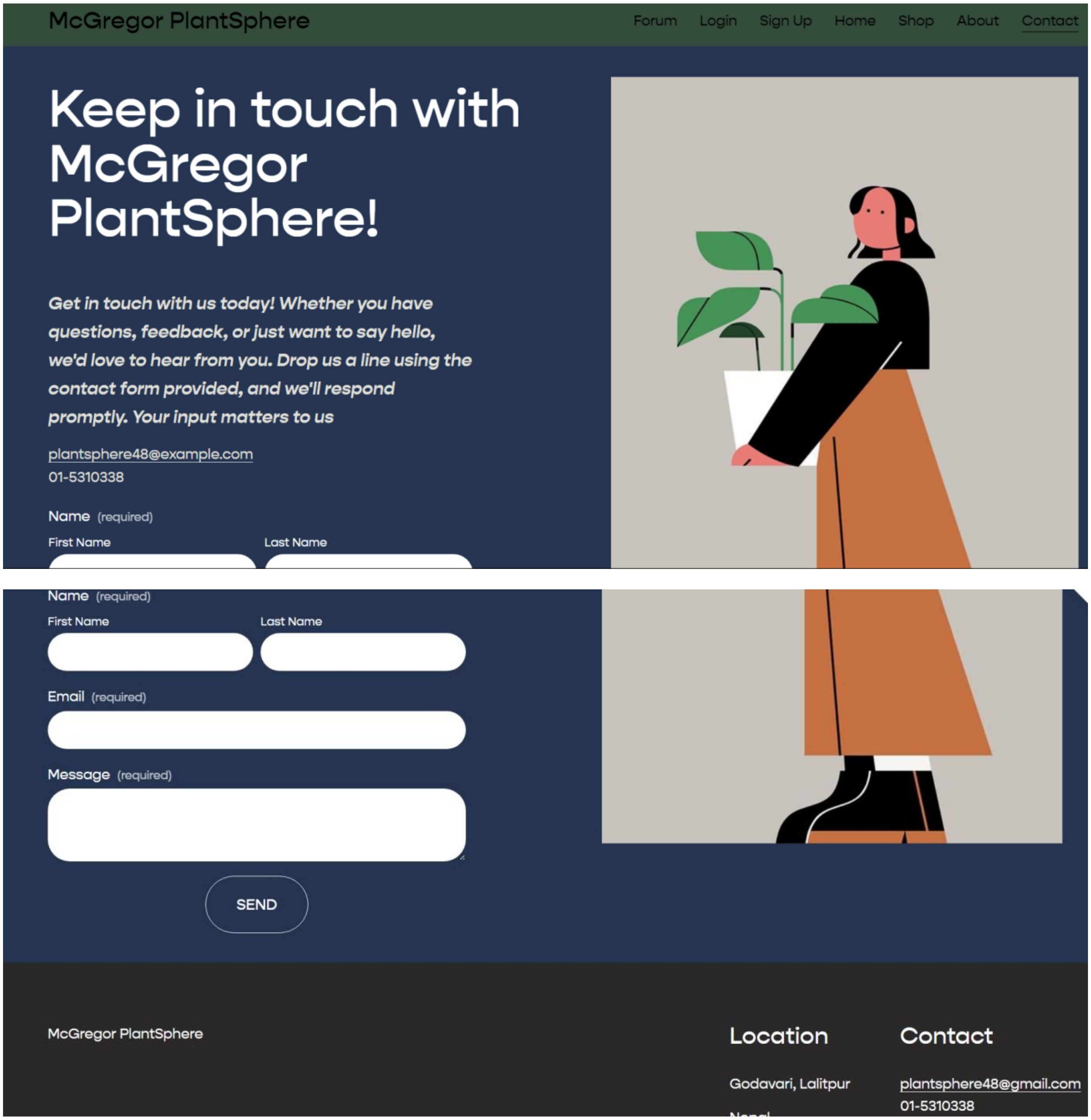
For over seven years, we've offered specialized undergraduate and postgraduate courses in agriculture and horticulture, affiliated with Dublin City University

building a community of plant enthusiasts. Our platform serves as a hub for discussions, conservation efforts, and sharing plant varieties at minimal or no cost.

Explore McGregor PlantSphere!

Figure 22:About Us Page

10.6.Contact Us Page:



Keep in touch with McGregor PlantSphere!

Get in touch with us today! Whether you have questions, feedback, or just want to say hello, we'd love to hear from you. Drop us a line using the contact form provided, and we'll respond promptly. Your input matters to us

plantsphere48@example.com
01-5310338

Name (required)

First Name Last Name

Name (required)

First Name Last Name

Email (required)

Message (required)

McGregor PlantSphere

Location

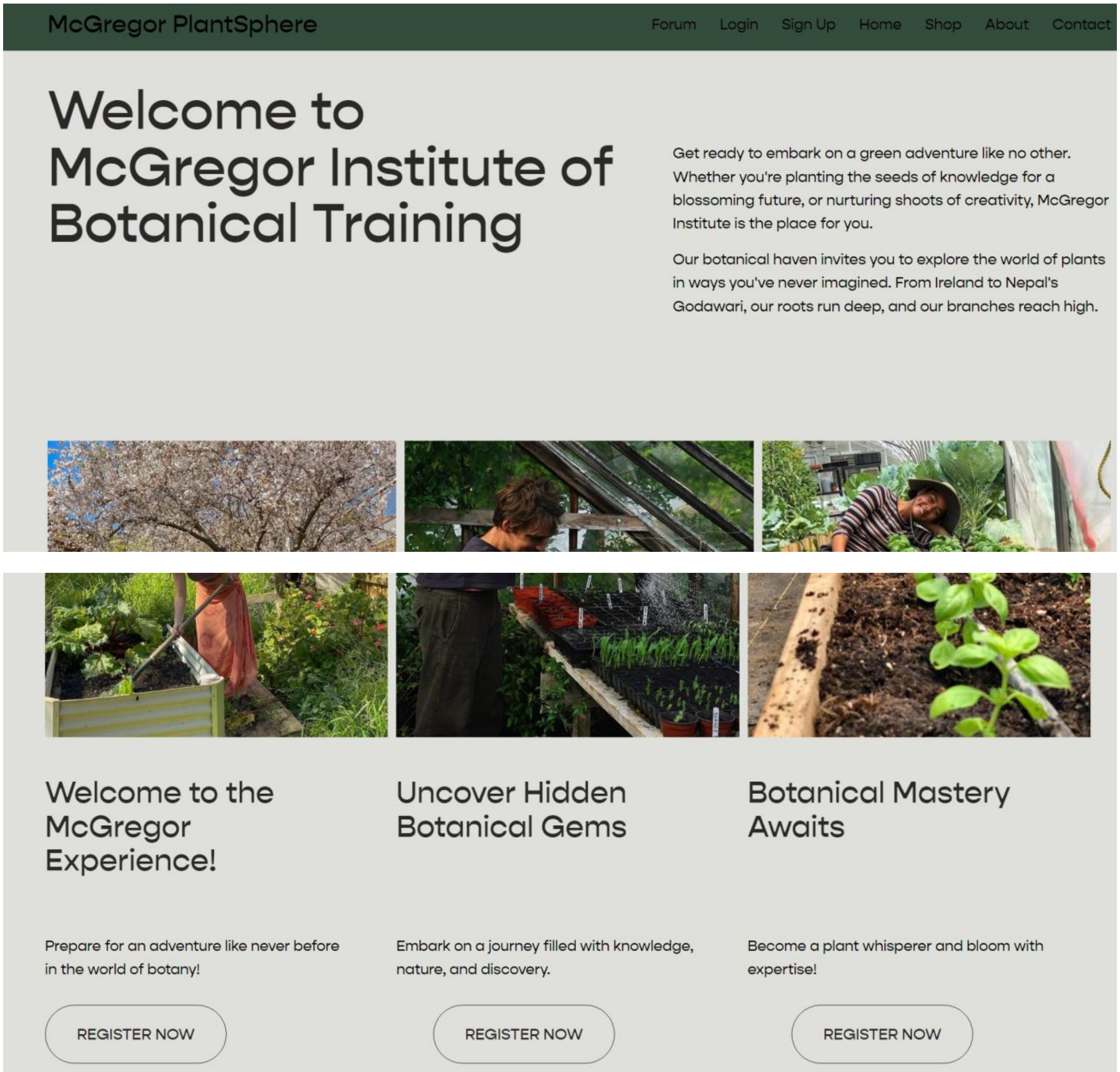
Godavari, Lalitpur
Nepal

Contact

plantsphere48@gmail.com
01-5310338

Figure 23: Contact Us Page

10.7. Our Services Page :



McGregor PlantSphere

Welcome to McGregor Institute of Botanical Training

Get ready to embark on a green adventure like no other. Whether you're planting the seeds of knowledge for a blossoming future, or nurturing shoots of creativity, McGregor Institute is the place for you.

Our botanical haven invites you to explore the world of plants in ways you've never imagined. From Ireland to Nepal's Godawari, our roots run deep, and our branches reach high.

Welcome to the McGregor Experience!

Prepare for an adventure like never before in the world of botany!

Uncover Hidden Botanical Gems

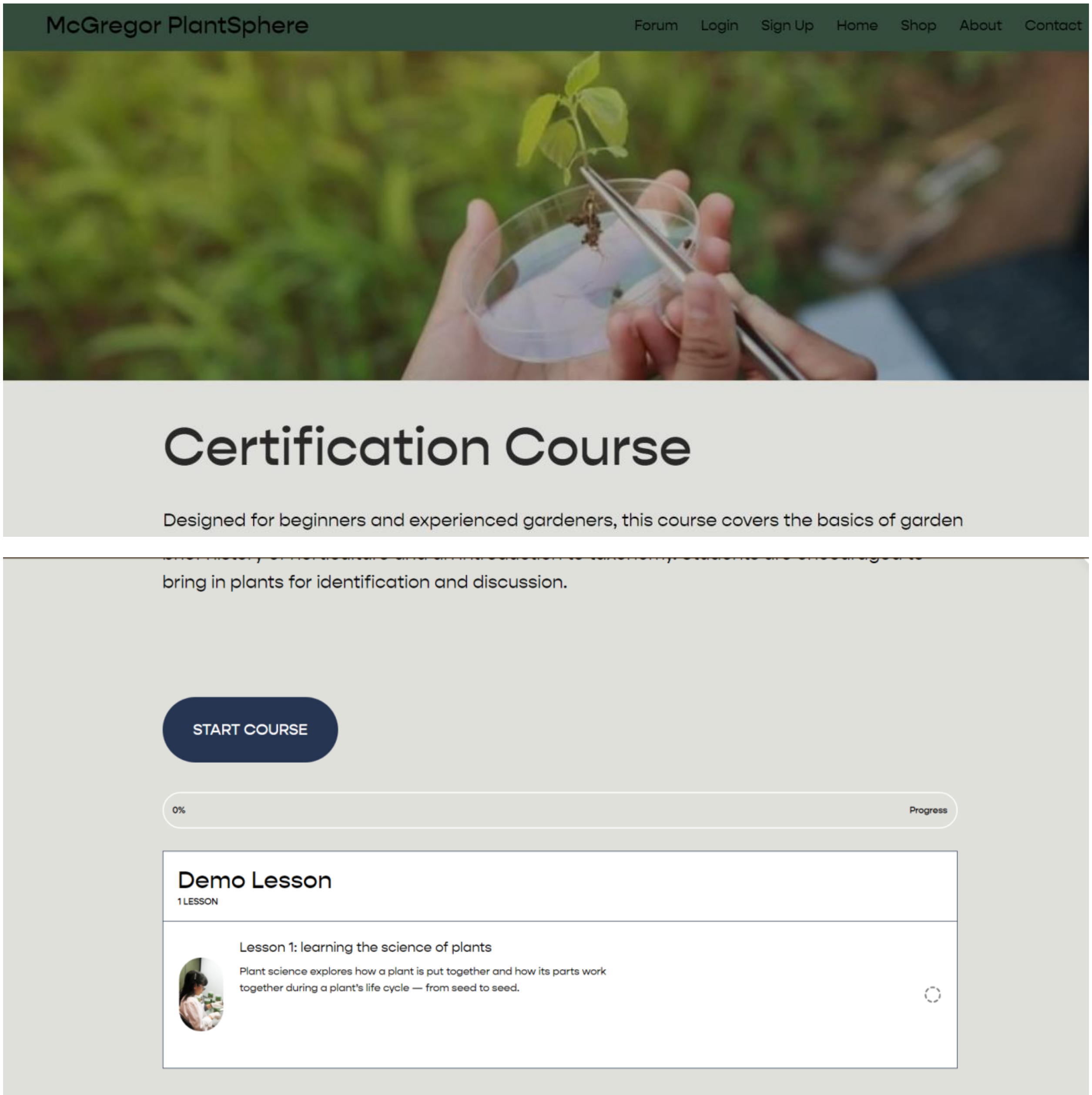
Embark on a journey filled with knowledge, nature, and discovery.

Botanical Mastery Awaits

Become a plant whisperer and bloom with expertise!

Figure 24: Our Service Page

10.8. Certification Course Page:



Certification Course

Designed for beginners and experienced gardeners, this course covers the basics of garden

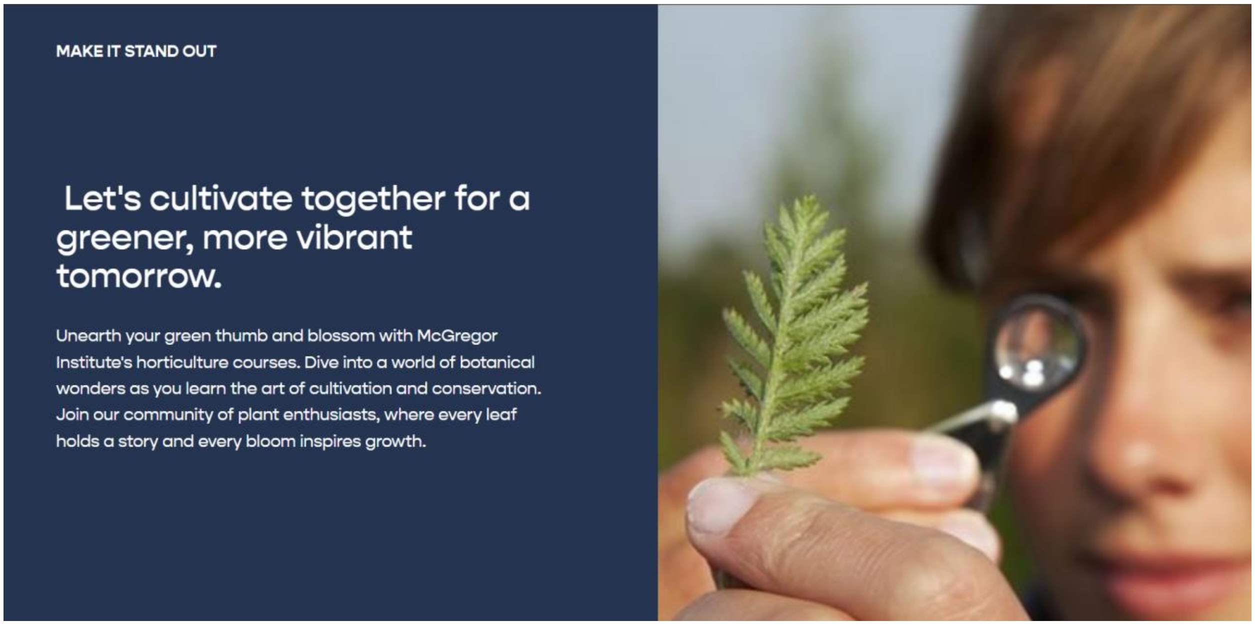
bring in plants for identification and discussion.

Demo Lesson

1 LESSON

Lesson 1: learning the science of plants

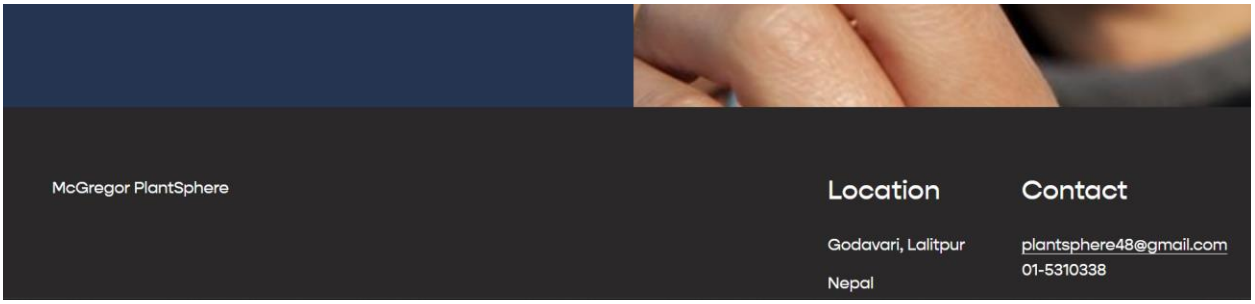
Plant science explores how a plant is put together and how its parts work together during a plant's life cycle — from seed to seed.



MAKE IT STAND OUT

Let's cultivate together for a greener, more vibrant tomorrow.

Unearth your green thumb and blossom with McGregor Institute's horticulture courses. Dive into a world of botanical wonders as you learn the art of cultivation and conservation. Join our community of plant enthusiasts, where every leaf holds a story and every bloom inspires growth.



McGregor PlantSphere

Location

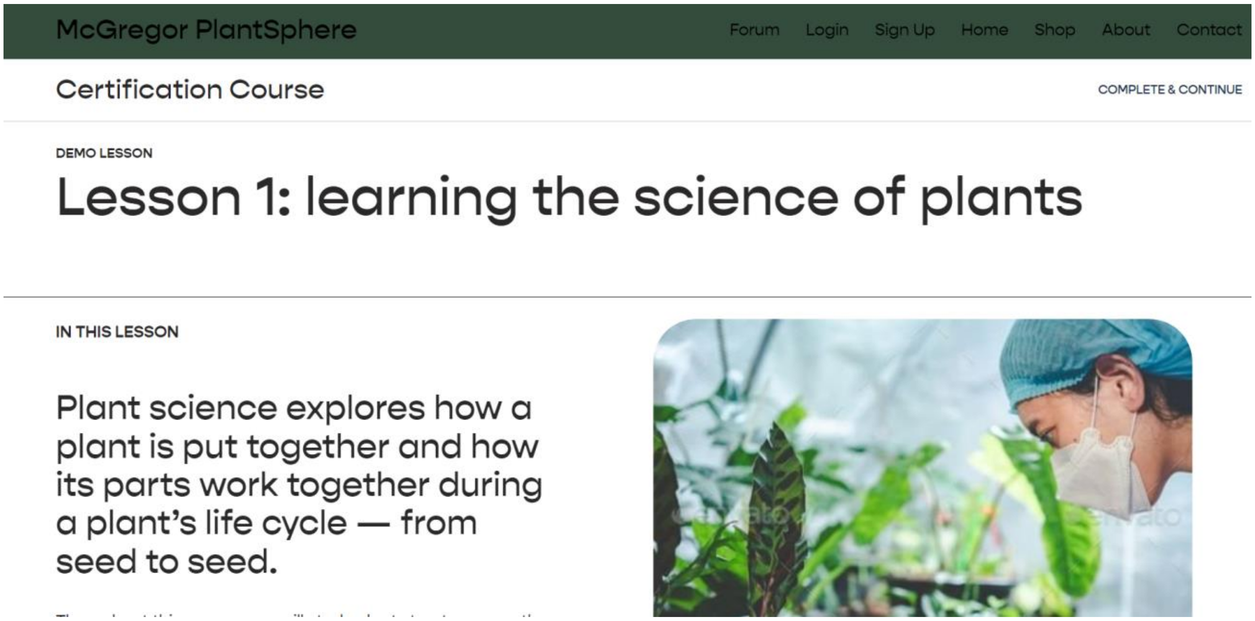
Godavari, Lalitpur

Nepal

Contact

plantsphere48@gmail.com

01-5310338



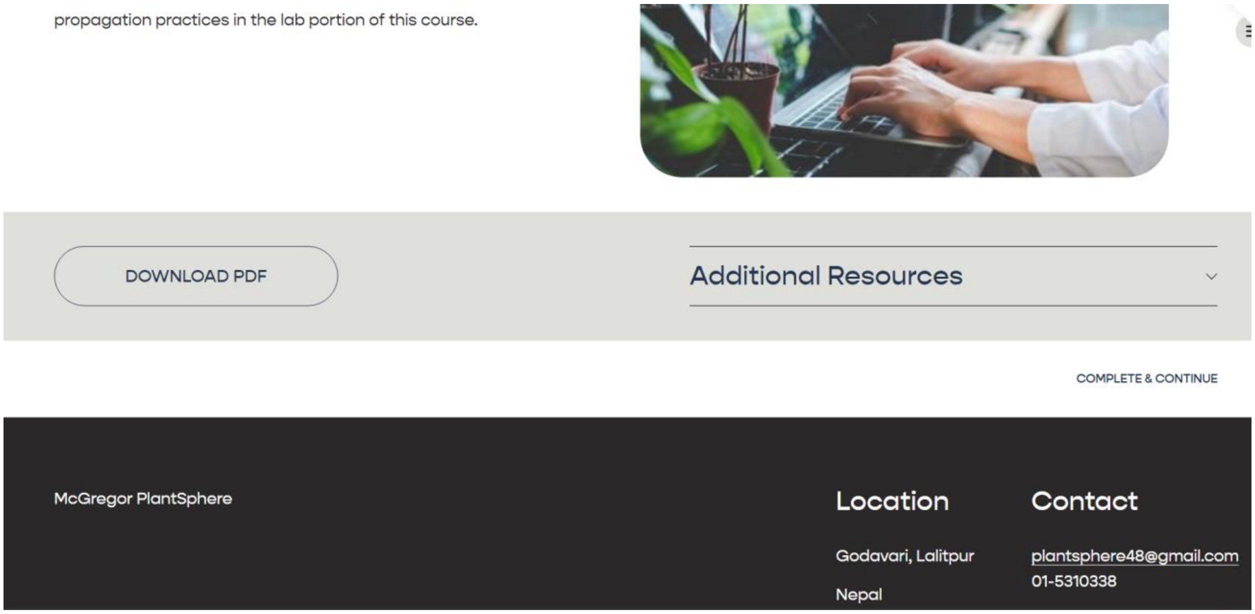
Certification Course

DEMO LESSON

Lesson 1: learning the science of plants

IN THIS LESSON

Plant science explores how a plant is put together and how its parts work together during a plant’s life cycle — from seed to seed.



propagation practices in the lab portion of this course.

McGregor PlantSphere

Location

Godavari, Lalitpur

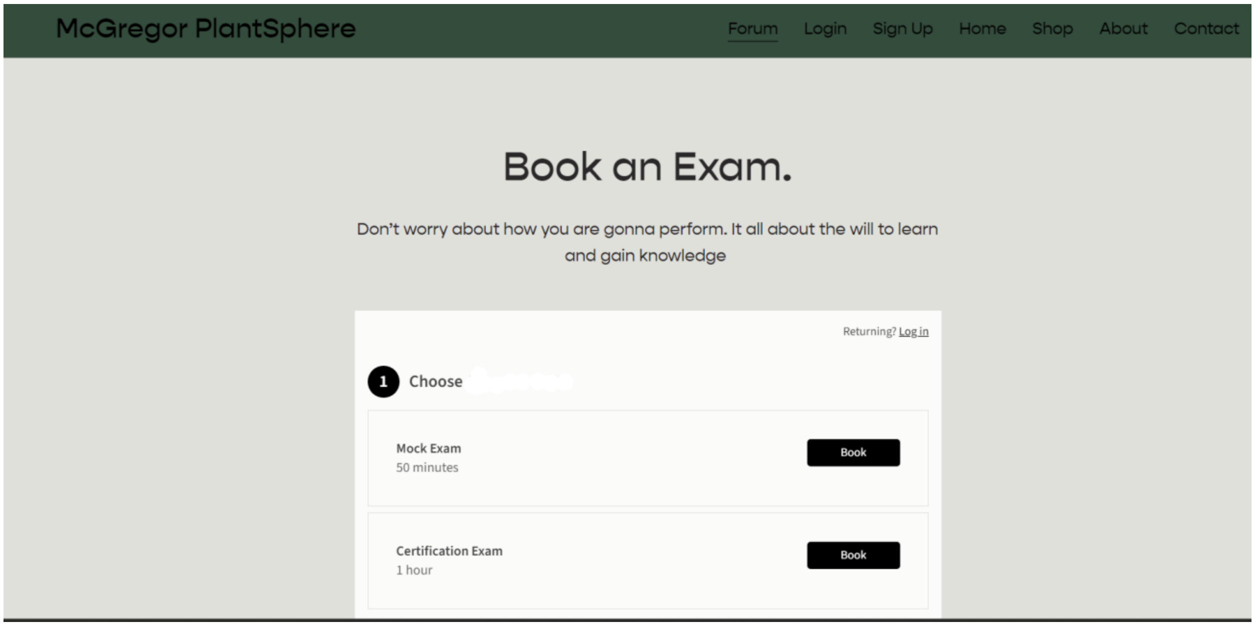
Nepal

Contact

plantsphere48@gmail.com
01-5310338

Figure 25: Certification Courses Page

10.9. Book Exam Page :



Book an Exam.

Don't worry about how you are gonna perform. It all about the will to learn and gain knowledge

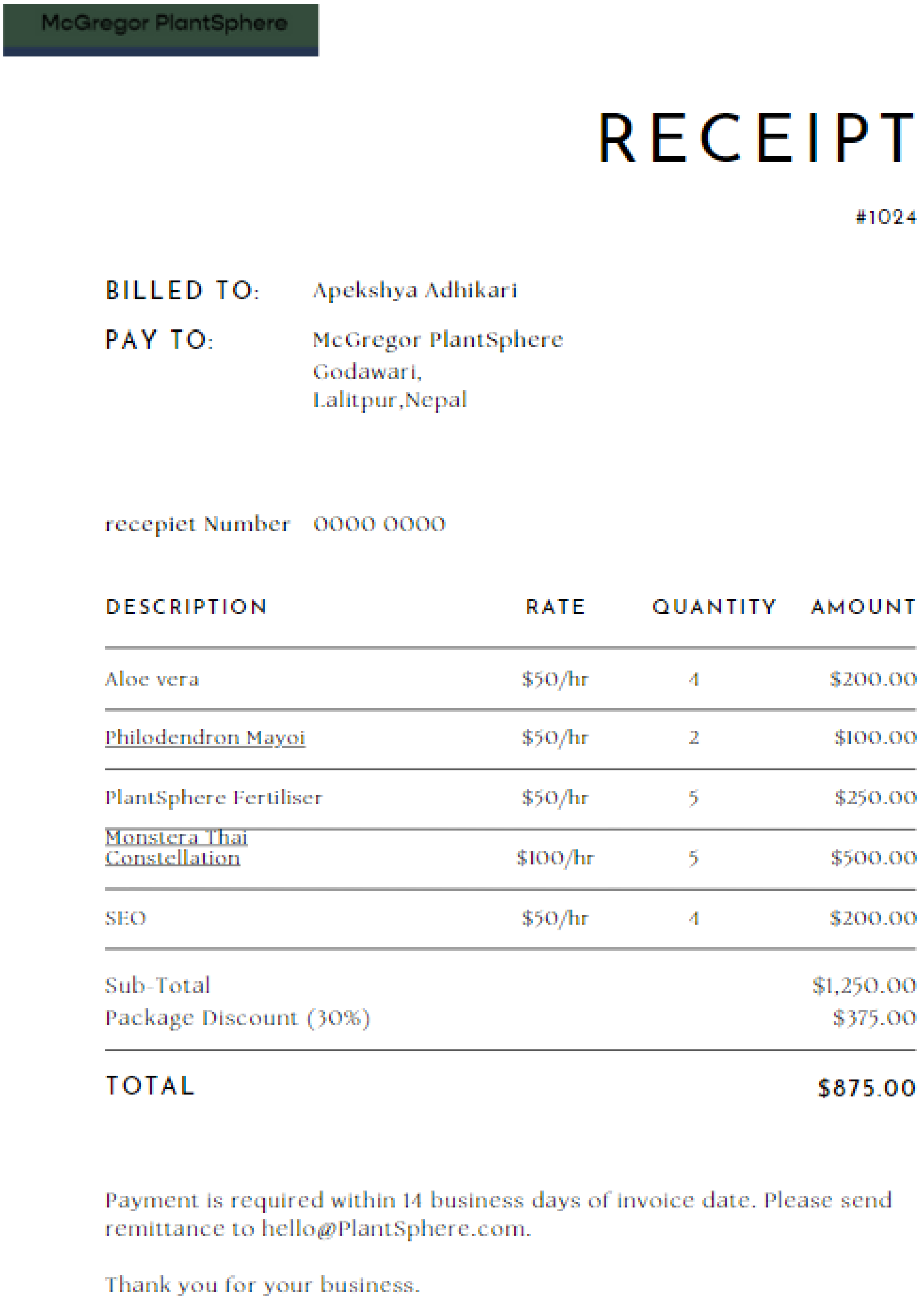
1 Choose

Mock Exam
50 minutes

Certification Exam
1 hour

Figure 26: Book Exam Page

10.10. Receipt Prototype:



McGregor PlantSphere

RECEIPT

#1024

BILLED TO: Apekshya Adhikari

PAY TO: McGregor PlantSphere
Godawari,
Lalitpur,Nepal

recepiet Number 0000 0000

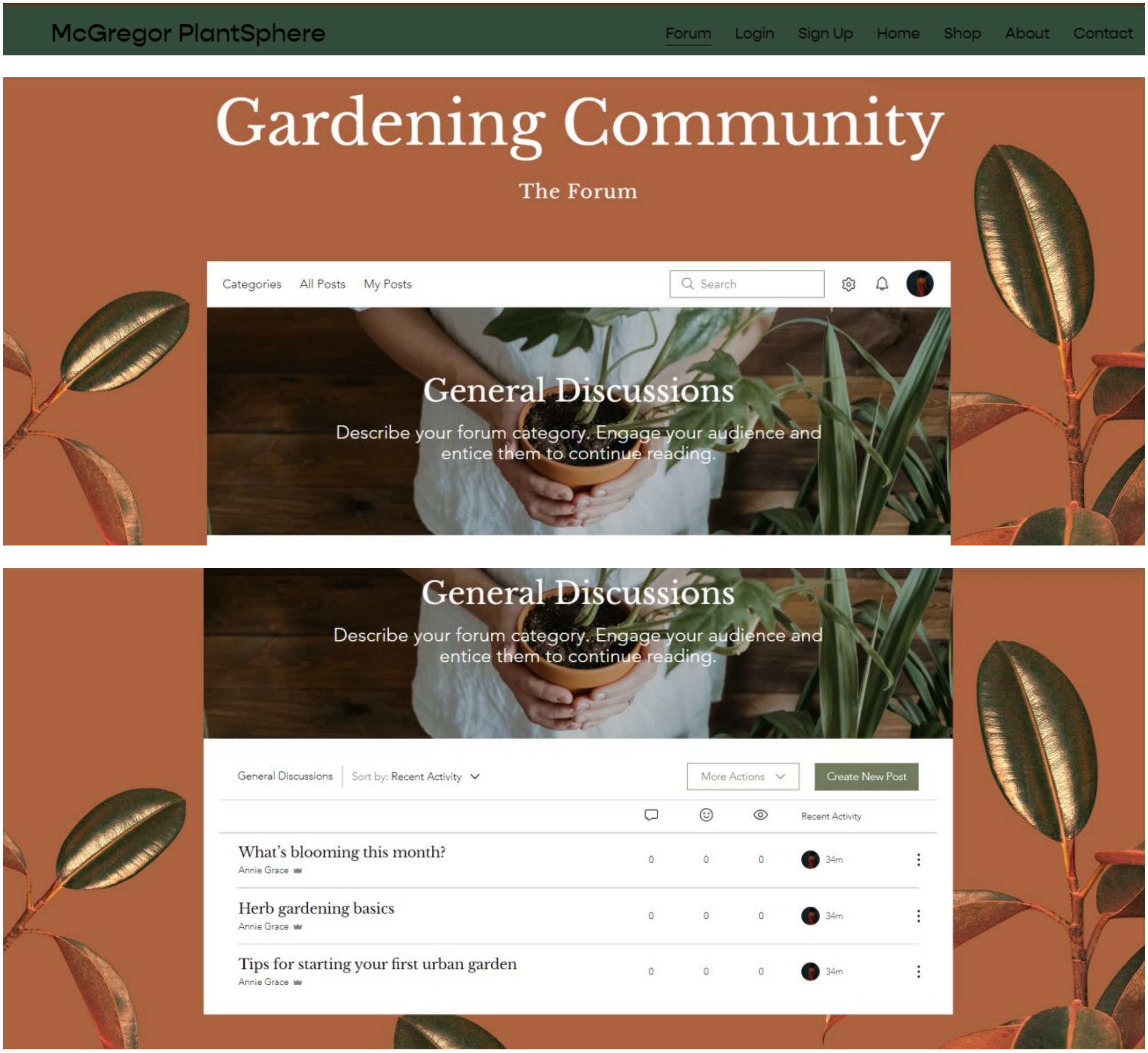
DESCRIPTION	RATE	QUANTITY	AMOUNT
Aloe vera	$50/hr	4	$200.00
Philodendron Mayoi	$50/hr	2	$100.00
PlantSphere Fertiliser	$50/hr	5	$250.00
Monstera Thai Constellation	$100/hr	5	$500.00
SEO	$50/hr	4	$200.00
Sub-Total			$1,250.00
Package Discount (30%)			$375.00
TOTAL			**$875.00**

Payment is required within 14 business days of invoice date. Please send remittance to hello@PlantSphere.com.

Thank you for your business.

Figure 27: Receipt Prototype

10.11: Forum Page:



Describe your forum category. Engage your audience and entice them to continue reading.

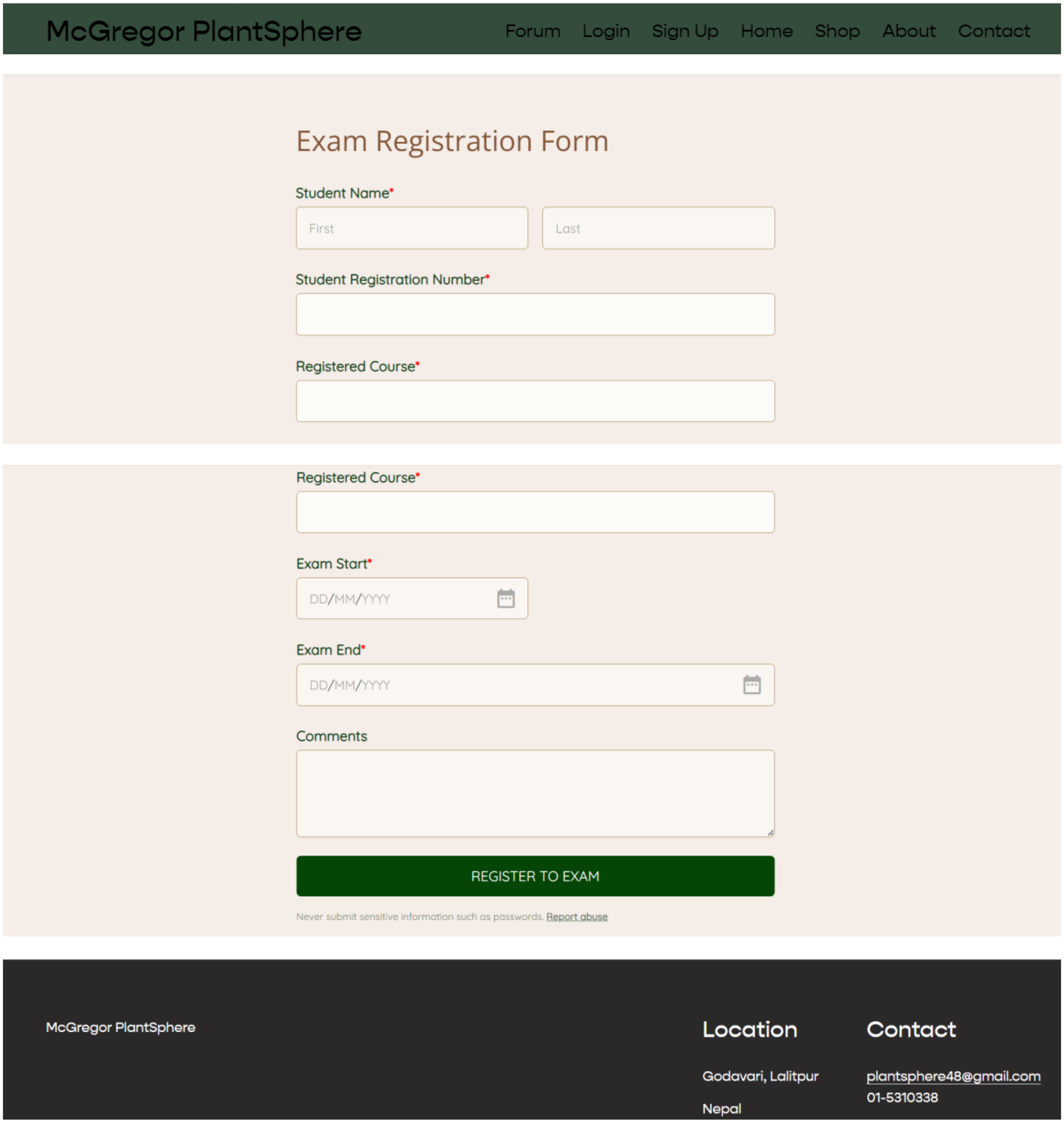
General Discussions

Describe your forum category. Engage your audience and entice them to continue reading.

	💬	☺	👁	Recent Activity
What's blooming this month? Annie Grace	0	0	0	34m
Herb gardening basics Annie Grace	0	0	0	34m
Tips for starting your first urban garden Annie Grace	0	0	0	34m

Figure 28: Forum Page

10.12. Exam Registration Form:



Exam Registration Form

Student Name*

First

Last

Student Registration Number*

Registered Course*

Registered Course*

Exam Start*

DD/MM/YYYY

Exam End*

DD/MM/YYYY

Comments

Never submit sensitive information such as passwords. Report abuse

McGregor PlantSphere

Location

Godavari, Lalitpur

Nepal

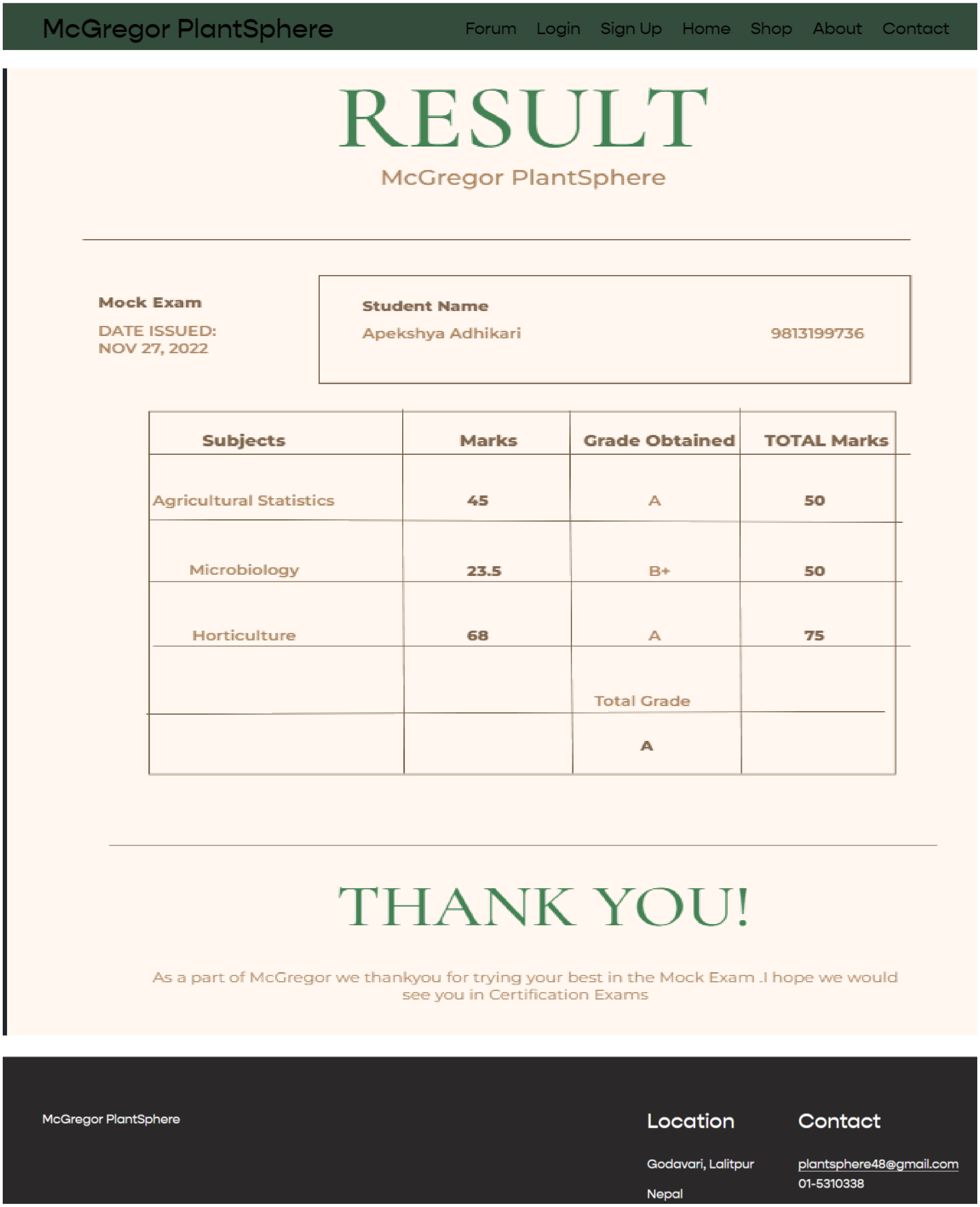
Contact

plantsphere48@gmail.com

01-5310338

Figure 29: Exam Registration Form

10.13.Result Prototype:



McGregor PlantSphere

RESULT

McGregor PlantSphere

Mock Exam

DATE ISSUED:
NOV 27, 2022

Student Name

Apekshya Adhikari 9813199736

Subjects	Marks	Grade Obtained	TOTAL Marks
Agricultural Statistics	45	A	50
Microbiology	23.5	B+	50
Horticulture	68	A	75
		Total Grade	
		A	

THANK YOU!

As a part of McGregor we thankyou for trying your best in the Mock Exam .I hope we would see you in Certification Exams

McGregor PlantSphere

Location

Godavari, Lalitpur

Nepal

Contact

plantsphere48@gmail.com
01-5310338

Figure 30: Result Prototype

References

Gantt, 2021. *Gantt.* [Online]
Available at: www.gantt.com
[Accessed 02 May 2024].

IBM Developer, n.d. *IBM Developer.* [Online]
Available at: https://developer.ibm.com/articles/the-sequence-diagram/

Inflectra Coorporation, 2024. *Inflectra.* [Online]
Available at: https://www.inflectra.com/Ideas/Topic/Use-Cases.aspx
[Accessed 04 May 2024].

Visual-Paradigm, 2022. *Visual Paradigm.* [Online]
Available at: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml-collaboration-diagram/
[Accessed 05 May 2024].

WorkBreakdownStructure, 2023. *Work Breadown Structure.* [Online]
Available at:
https://www.workbreakdownstructure.com/#:~:text=The%20Work%20Breakdown%20Structure%20is,schedule%20and%20cost%20estimate%20activities.