DATABASE DESIGN

Database Design is the process of structuring and organizing data in a database to ensure efficient storage, retrieval, and management. It involves defining tables, relationships, constraints, and indexing strategies to optimize performance and maintain data integrity.

Database tables:

1. Admin

Field Name	Type	Size	Constraints	Description
Admin id	Int	15	Primary Key	Identification number of admin
Admin Username	Varchar	50	Not Null	Name of admin
Admin Email	Varchar	50	Not Null	Email address of admin
Admin Password	Varchar	15	Not Null	Password of admin

2. Category

Field Name	Type	Size	Constraints	Description
Category id	Int	15	Primary Key	Identification number of category
Category Name	Varchar	50	Not Null	Name of category
Category Description	Varchar	50	Not Null	Description of category
Created Date	DateTime	15		Category added date

3. Care Tip

Field Name	Type	Size	Constraints	Description
Care Tip id	Int	15	Primary Key	Identification number of care tips
Category id	Int	15	Foreign Key	Identification number of categories
Product id	Int	15	Foreign Key	Identification number of products
Title	Varchar	50	Not Null	Title of care tips
Description	Varchar	50		Description of care tips
Created Date	Date	15		Care tips added date

4. Nursery

Field Name	Type	Size	Constraints	Description
Nursery id	Int	15	Primary Key	Identification number of nursery
Nursery Name	Varchar	50	Not Null	Name of nursery
Location	Varchar	15		Location of nursery
Address	Varchar	50		Address of nursery
Phone	Int	12		Phone number of nurseries
Owner Name	Varchar	50		Name of the owner of nursery
License Number	Varchar	15		License number of nursery
Email	Varchar	50		Email address of nursery
Password	Varchar	15		Password of the nursery
Status	Int	15		Status of nursery approved/not

5. Product

Field Name	Type	Size	Constraints	Description
Product id	Int	15	Primary Key	Identification number of nursery
Nursery id	Int	15	Foreign Key	Identification number of nursery
Category id	Int	15	Foreign Key	Identification number of category
Product Name	Varchar	50		Name of the product
Description	Varchar	50		Description of the product
Price	Varchar	50		Price of the product
Image	Varchar	15		Image of the product
Created Date	Date	15		Product added date
Stock	Int	15		Number of available products

6. User

Field Name	Type	Size	Constraints	Description
User id	Int	15	Primary Key	Identification number of user
User Name	Int	15		Name of the user
Address	Int	15		Address of the user
Gender	Varchar	50		Gender of the user
Location	Varchar	50		Location of the user
Email	Varchar	50		Email address of the user
Password	Varchar	15		Password of the user
Phone	Int	12		Phone number of the user

7. Booking

Field Name	Type	Size	Constraints	Description
Booking id	Int	15	Primary Key	Identification number of booking
User id	Int	15	Foreign Key	Identification number of user
Product id	Int	15	Foreign Key	Identification number of product
Booking Quantity	Int	50		Number of products booked
Booking Amount	Int	50		Total amount of booking
Booking Date	Date	50		Date of booking
Status	Varchar	15		Status of booking

8. Cart

Field Name	Type	Size	Constraints	Description
Cart id	Int	15	Primary Key	Identification number of nursery
Product id	Int	15	Foreign Key	Identification number of nursery
Status	Int	15		Status of the product in the cart

UI DESIGN

Under the UI Design section, showcasing screenshots of the user interface enhances the project's visual presentation and provides users with a glimpse of the application's layout and functionality. By including images users can visually understand the navigation flow and design aesthetics. These screenshots offer a quick overview of the application's appearance and enable users to familiarize themselves with its interface.

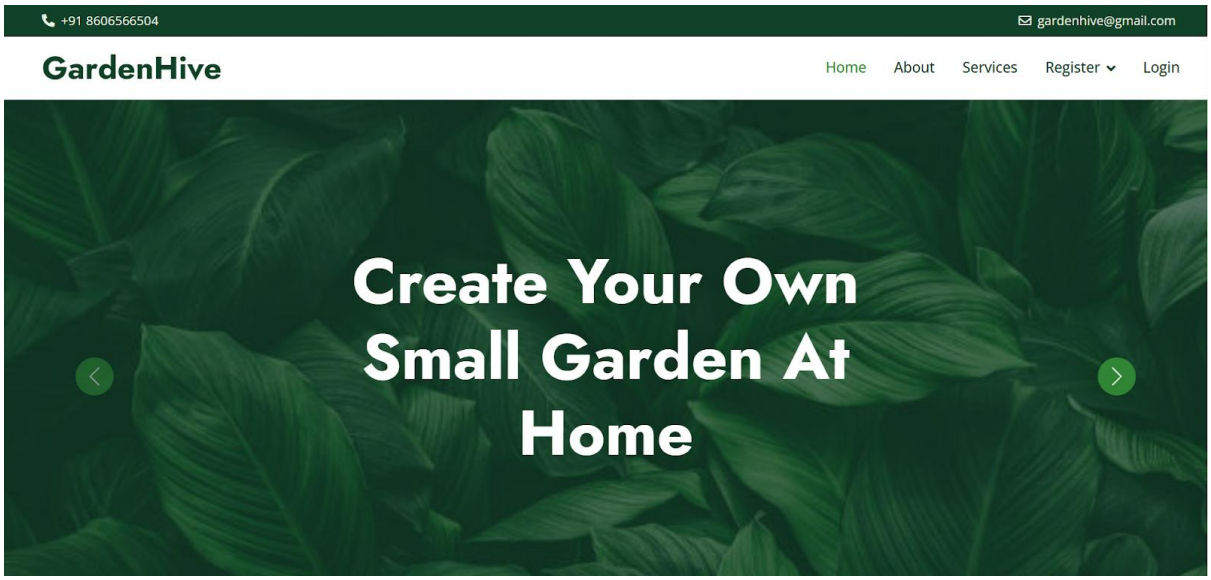

Figure 13: Index Page

Figure 13 Index page is every first page that a user views when enters into the website. From this page user can visit about, services, new nurseries and users can register and login into the system.

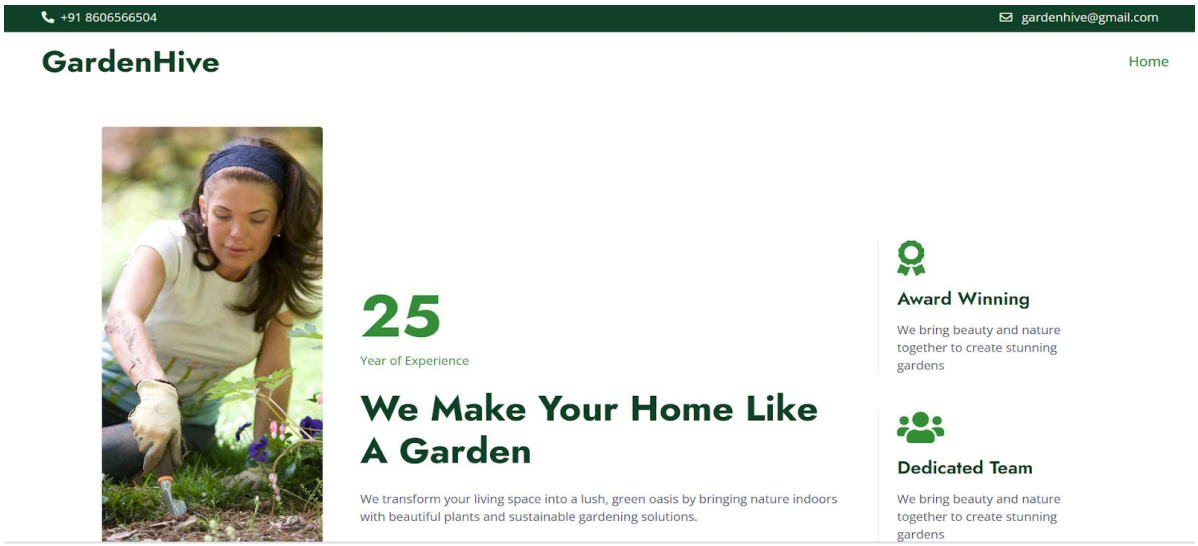

Figure 14: About Page

Figure 14 shows about page which describes about Garden Hive, its years of experience, awards achieved etc.

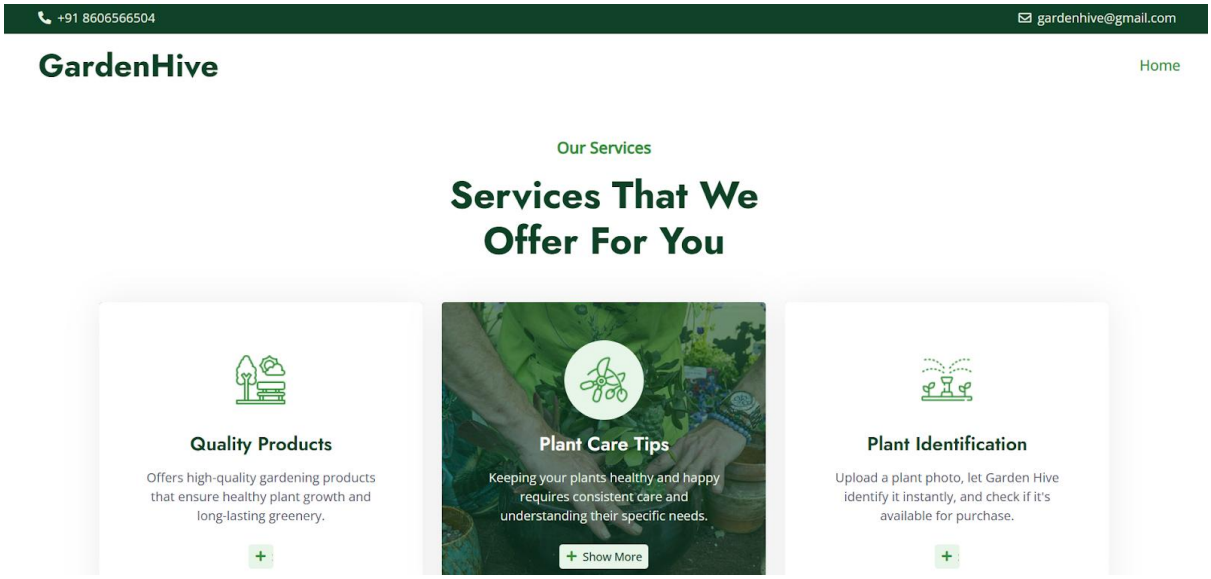

Figure 15: Service Page

Figure 15 Service page shows the services provided by Garden Hive. Users can view details by login to the system.

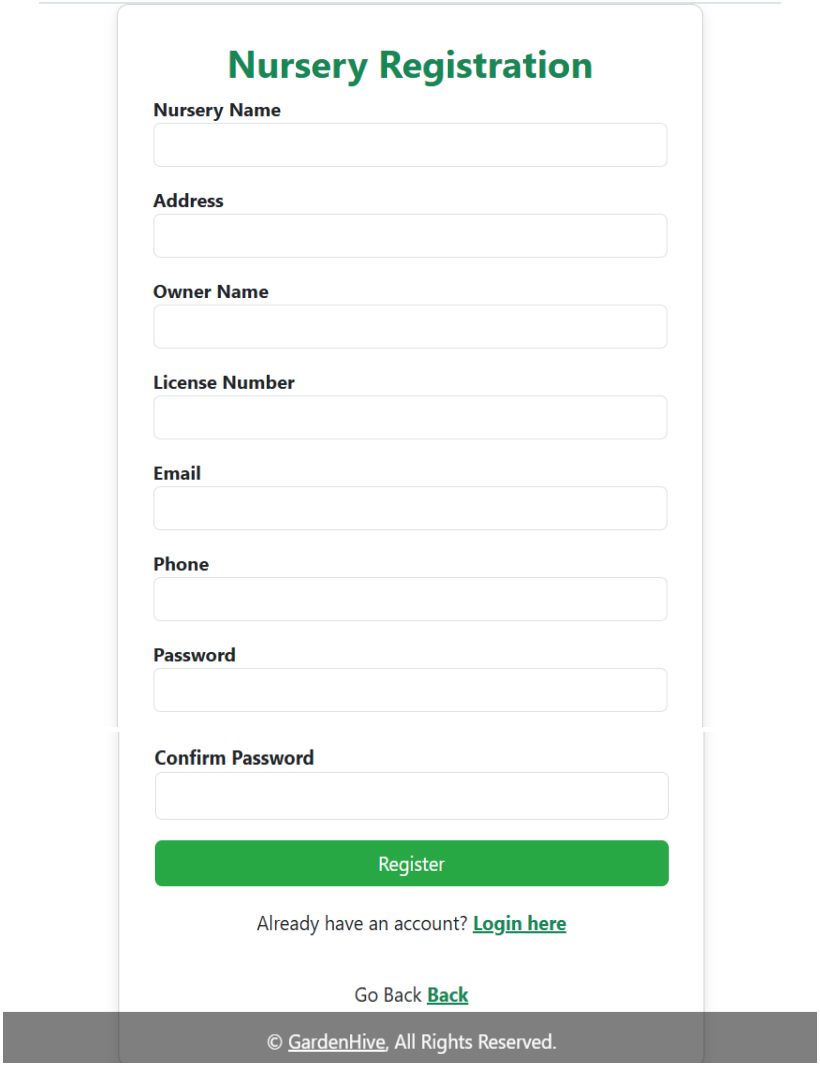

Figure 16: Nursery Registration Page

Figure 16 Nursery registration page, here nurseries can register by filling the fields. The registration will be verified by admin. Admin verified nurseries can only login to the system and utilize the functionalities.

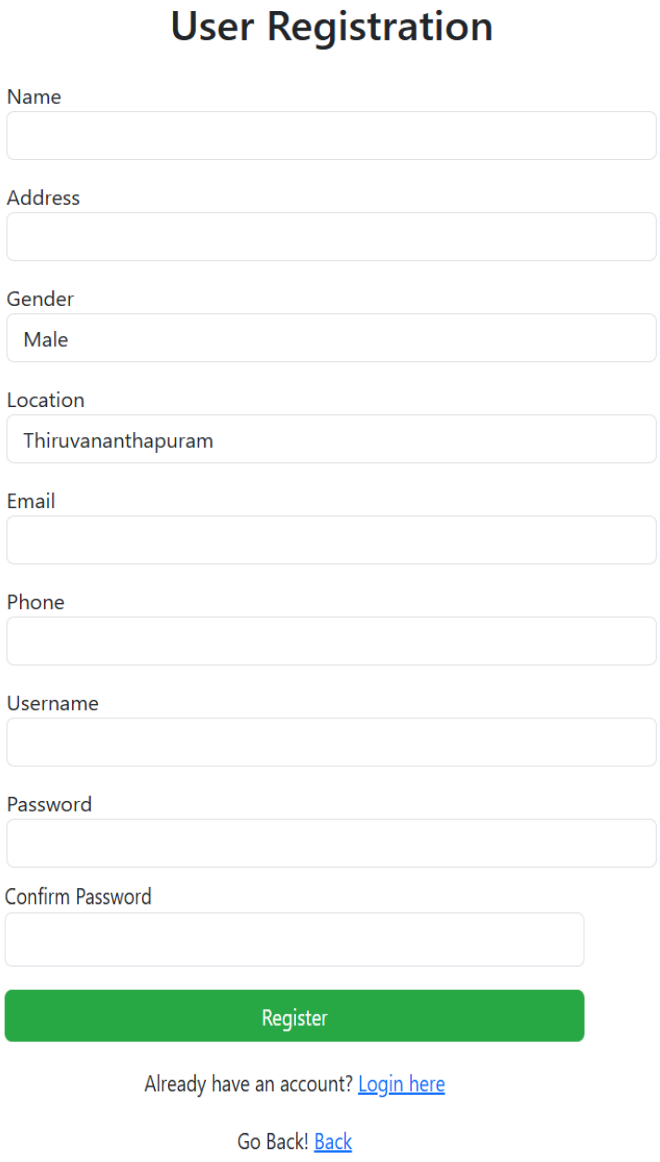

Figure 17: User Registration Page

Figure 17 User Registration Page, here users can register, registered users can only view products, book and add products to cart.

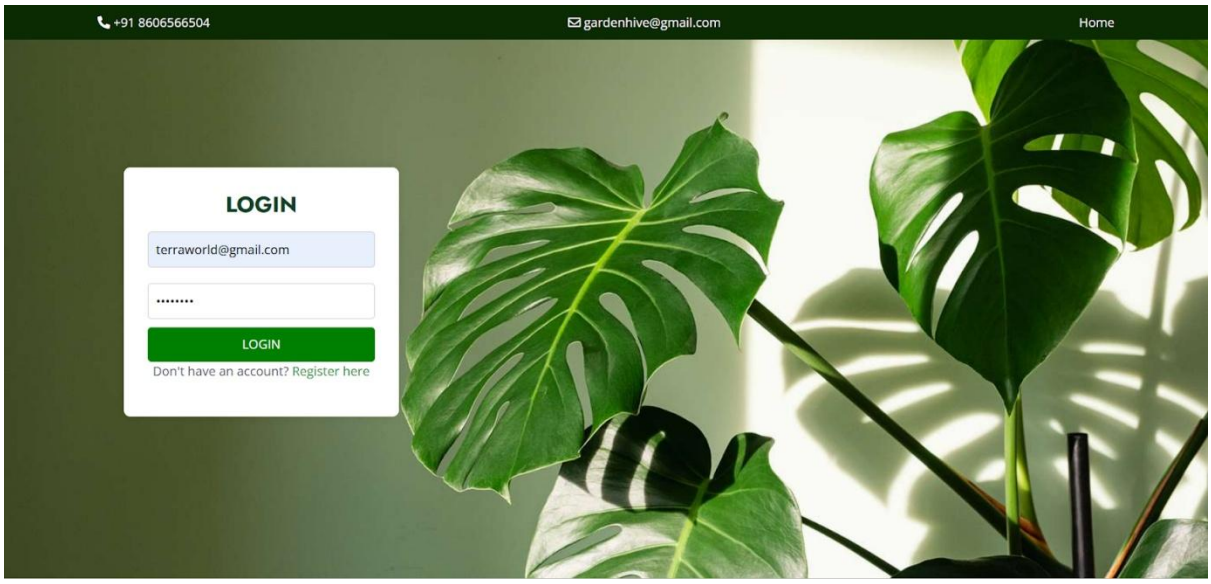

Figure 18: Login Page

Figure 18 shows login page, admin, registered nurseries and users can login to the system.

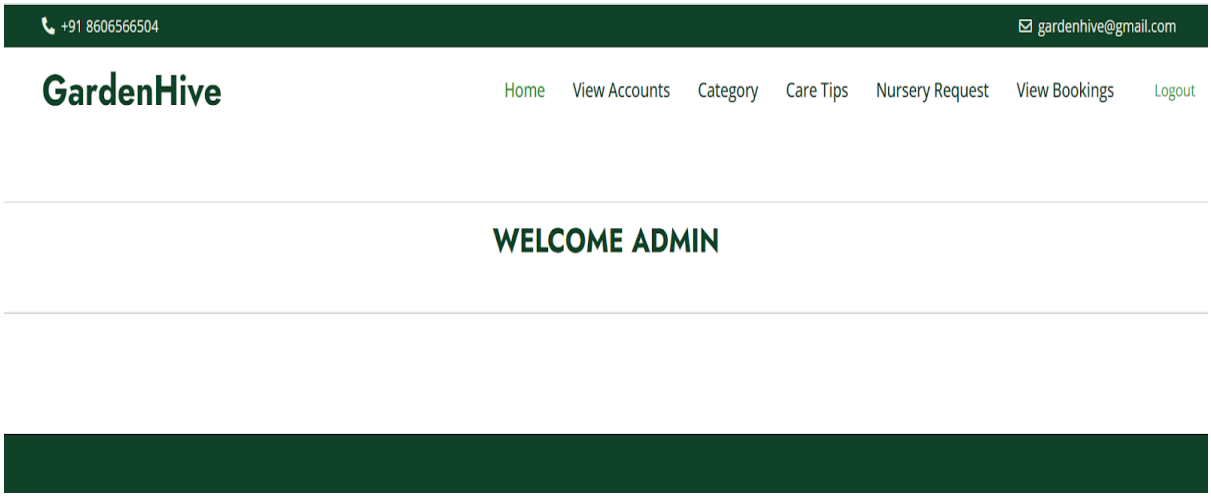

Figure 19: Admin Home Page

Figure 19 shows the admin home page when admin login it is directed to this page. Here admin can add category, care tips, approve nursery requests, manage users and nurseries, view bookings etc.

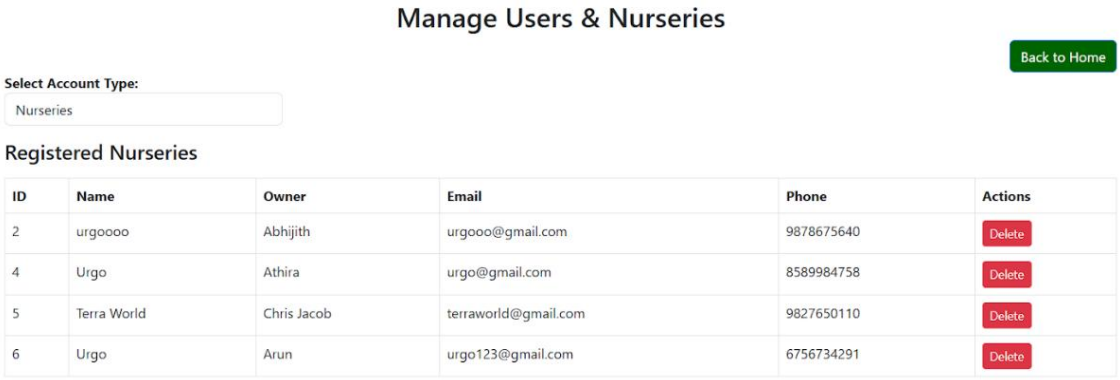

Figure 20: Admin Manage Nurseries

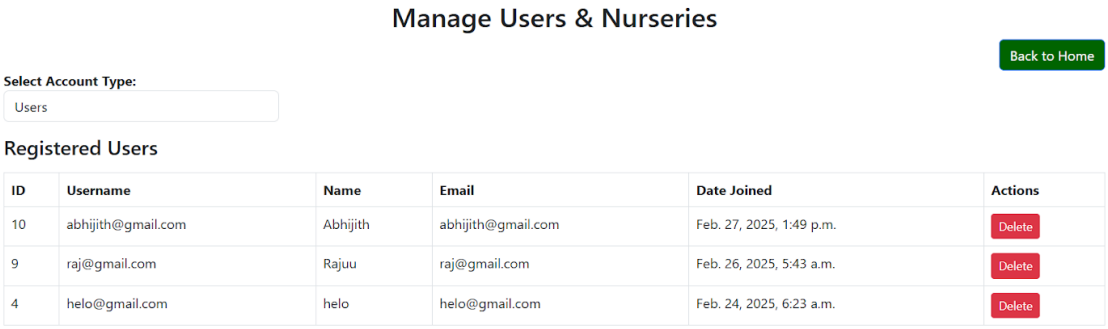

Figure 21: Admin Manage Users

Figure 20 and Figure 21 shows the page in which admin manages users and nurseries. Admin can view them if any unseen circumstances are seen admin can delete/remove nurseries and users.

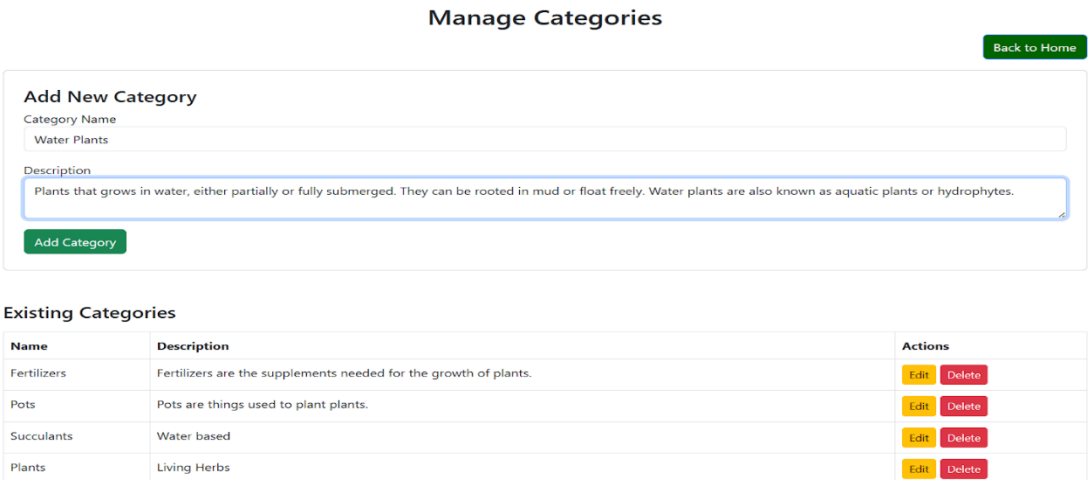

Figure 22: Admin Add Category

Figure 22 shows the page which allow the admin to add new categories. Admin can delete or edit the existing categories.

Figure 23: Admin Add Care tips

Figure 23 shows the page to add new care tips. Admin can add new care tips and also can delete or edit existing care tips.

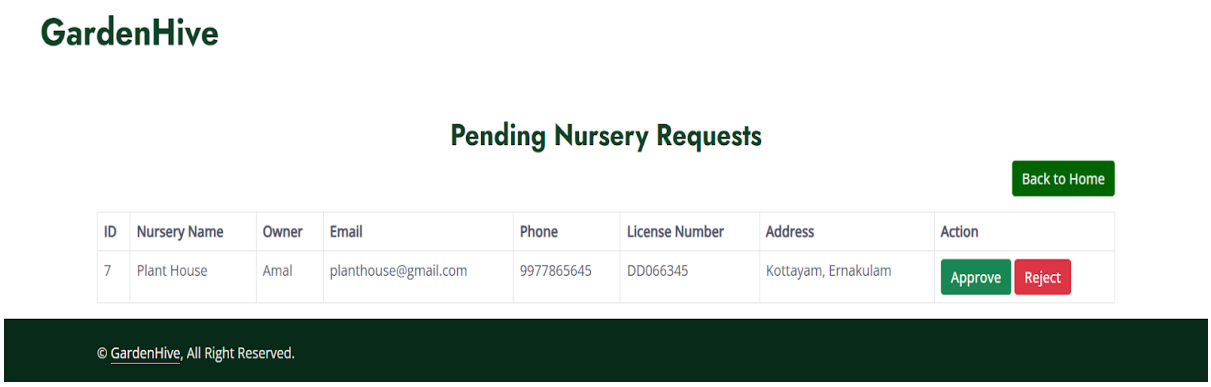

Figure 24: Admin Approve Nursery

Figure 24 shows the page which is able to approve the nursery registration. Admin can approve or reject the nursery registrations.

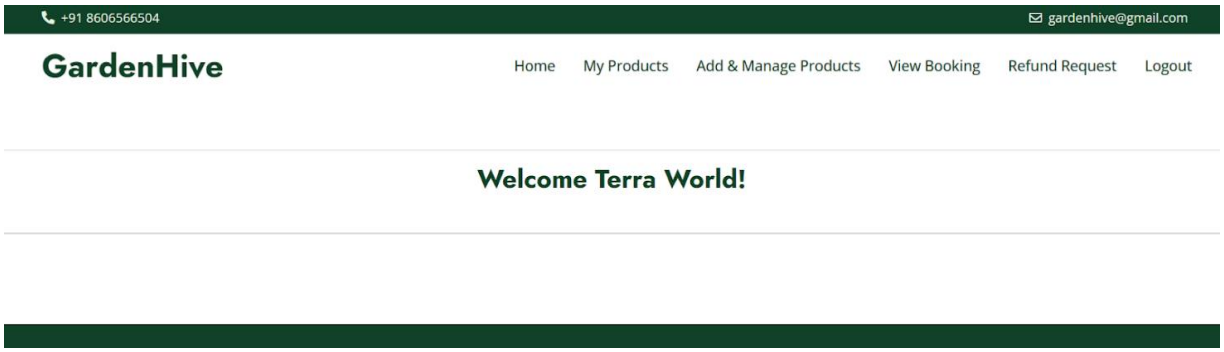

Figure 25: Nursery Home Page

Figure 25 shows the nursery home page. A registered nursery can add their products and also, they can manage the products.

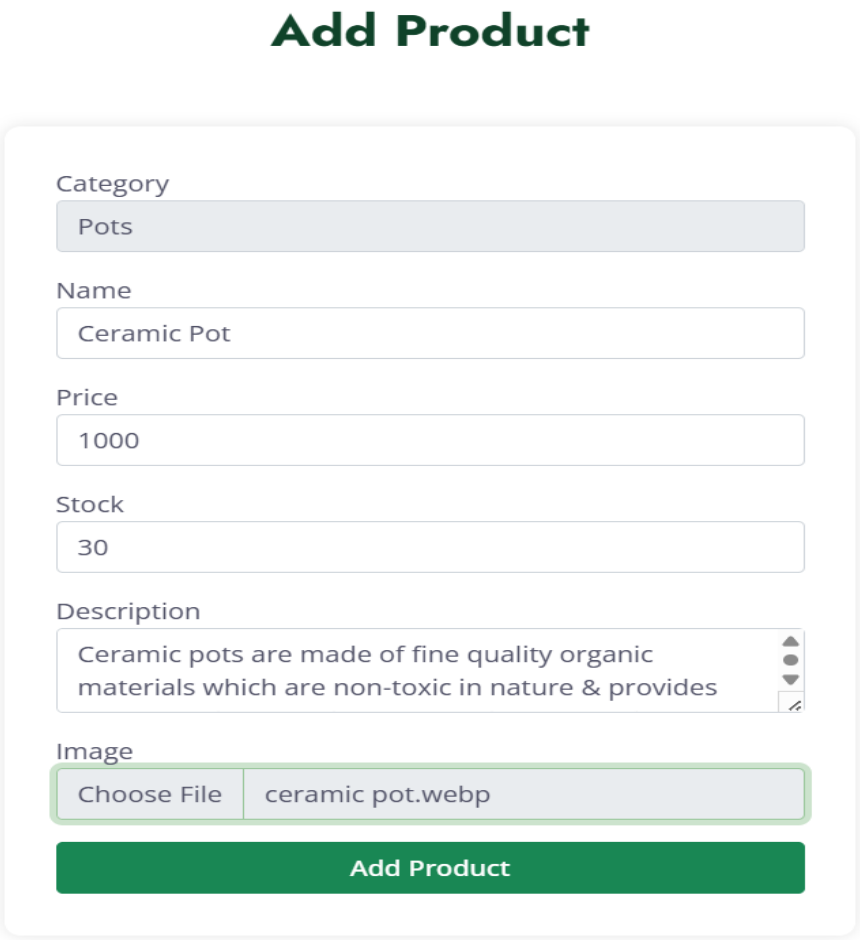

Figure 26: Add Product

Figure 26 shows the page through which nursery add new products.

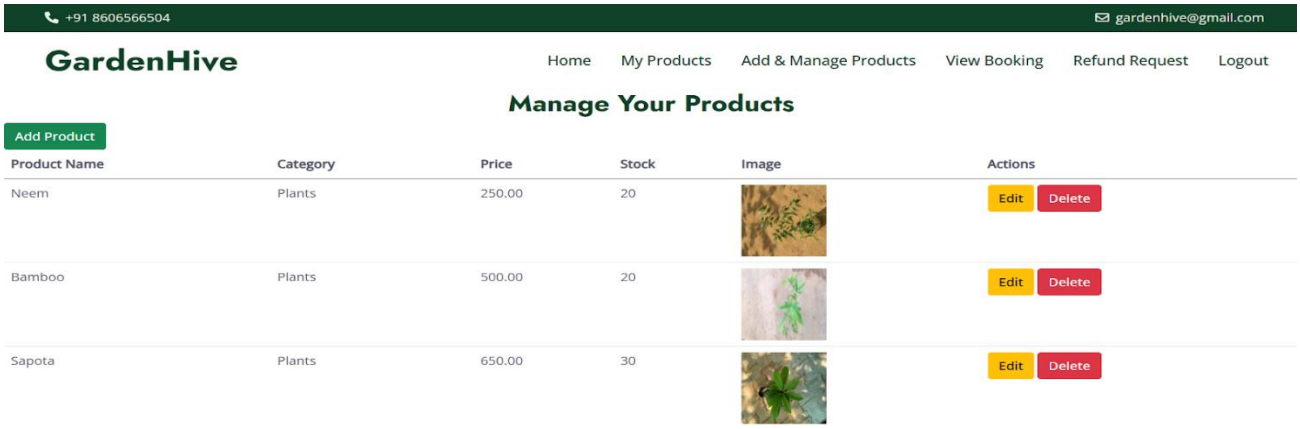

Figure 27: Manage Products

Figure 27 shows the page nursery manages the products. The nursery can edit or delete the existing product, nurseries can update stocks etc.

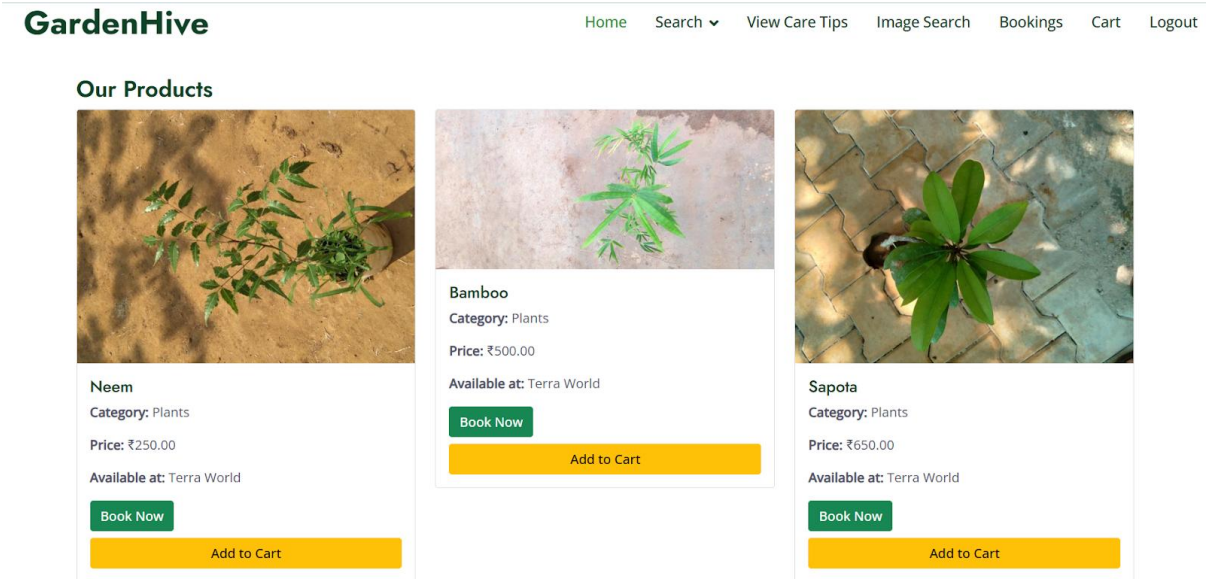

Figure 28: User Home Page

Figure 28 shows the user home page, users can search, view care tips, products and can book products.

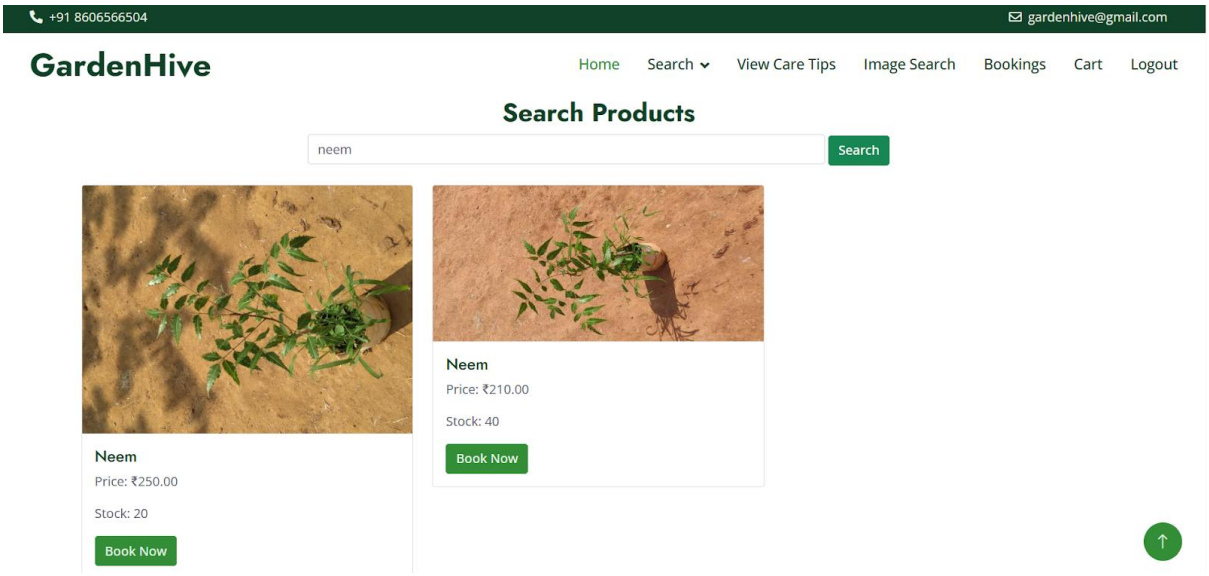

Figure 29: Search Product

Figure 29 shows the page which allow users to search products by its name and user can book these products.

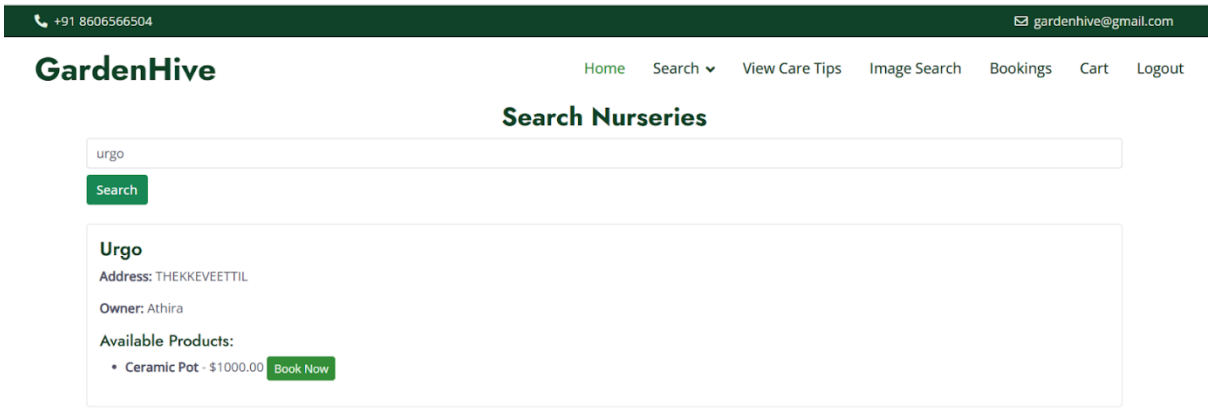

Figure 30: Search Nurseries

Figure 30 shows the page which allow users to search particular nursery and view their products and book if needed.

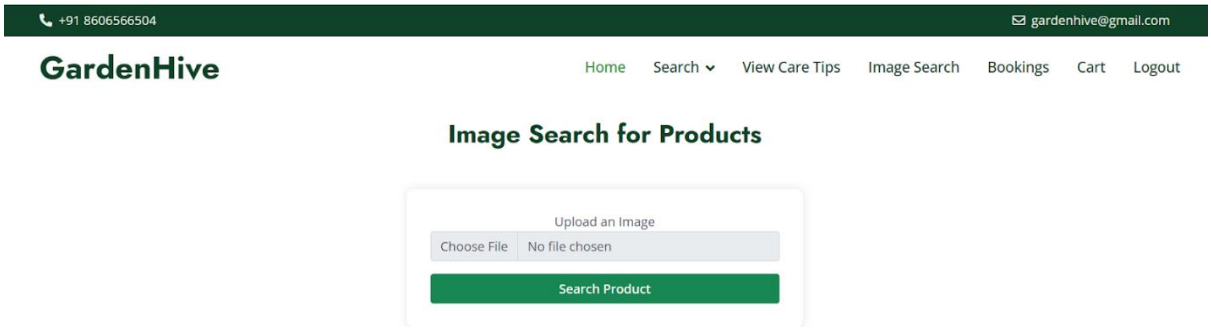

Figure 31: Image Search

Figure 31 shows the page to upload images of product and search for the products. If available product is searched it will be viewed by users and users can purchase according to their needs.

SYSTEM DESIGN

System design in the context of a deep learning project involves planning and structuring the components and workflows essential for building, training, and deploying deep learning models.

Model Building

Model building in the context of a deep learning project refers to the process of creating and training a neural network architecture to perform a specific task, such as image classification, object detection, or natural language processing. This process involves selecting an appropriate neural network architecture, defining its structure and parameters, preparing the training data, optimizing the model's performance through iterative training, and evaluating its performance on validation or test datasets. Model building is a crucial step in developing deep learning solutions and requires expertise in neural network design, optimization techniques, and domain-specific knowledge.

Model Planning

Model planning in the context of a deep learning project involves strategizing and outlining the approach for selecting, building, and optimizing the deep learning models to address a specific problem or task. In this project, we are using CNN architecture, model planning for the CNN model involves strategizing how to leverage the convolutional layers of CNN and adapting its fully connected layers to suit the specific requirements of the target task, ensuring effective transfer learning for optimal performance.

```
# Build the CNN model
model = Sequential([

    Conv2D(32, (3,3), activation='relu', input_shape=(224, 224, 3)),
    MaxPooling2D(2,2),

    Conv2D(64, (3,3), activation='relu'),
    MaxPooling2D(2,2),

    Conv2D(128, (3,3), activation='relu'),
    MaxPooling2D(2,2),

    Flatten(),

    Dense(256, activation='relu'),
    Dropout(0.5),
    Dense(num_classes, activation='softmax')
])

model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])

model.summary()
```

Figure 31: Snapshot of importing CNN model

Model: "sequential_4"

Layer (type)	Output Shape	Param #
conv2d_12 (Conv2D)	(None, 222, 222, 32)	896
max_pooling2d_12 (MaxPooling2D)	(None, 111, 111, 32)	0
conv2d_13 (Conv2D)	(None, 109, 109, 64)	18,496
max_pooling2d_13 (MaxPooling2D)	(None, 54, 54, 64)	0
conv2d_14 (Conv2D)	(None, 52, 52, 128)	73,856
max_pooling2d_14 (MaxPooling2D)	(None, 26, 26, 128)	0
flatten_4 (Flatten)	(None, 86528)	0
dense_8 (Dense)	(None, 256)	22,151,424
dropout_4 (Dropout)	(None, 256)	0
dense_9 (Dense)	(None, 10)	2,570

Total params: 22,247,242 (84.87 MB)
Trainable params: 22,247,242 (84.87 MB)
Non-trainable params: 0 (0.00 B)

Figure 32: Model Summary

Figure 32 shows the snapshot showing the model summary of CNN.

Model Training

Model training in deep learning involves feeding data through a neural network and updating its parameters iteratively to minimize the difference between the model's predictions and the actual target values.

```
dataset_path = "/content/drive/MyDrive/Plant Dataset"

datagen = ImageDataGenerator(
    rescale=1./255,
    validation_split=0.2
)

train_generator = datagen.flow_from_directory(
    dataset_path + "/Train",
    target_size=IMG_SIZE,
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    subset='training'
)

valid_generator = datagen.flow_from_directory(
    dataset_path + "/Valid",
    target_size=IMG_SIZE,
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    subset='validation'
)
```

Figure 33: Train the Model

```
EPOCHS = 20

history = model.fit(
    train_generator,
    validation_data=valid_generator,
    epochs=EPOCHS,
    verbose=1
)
```

Figure 34: Compile the Model

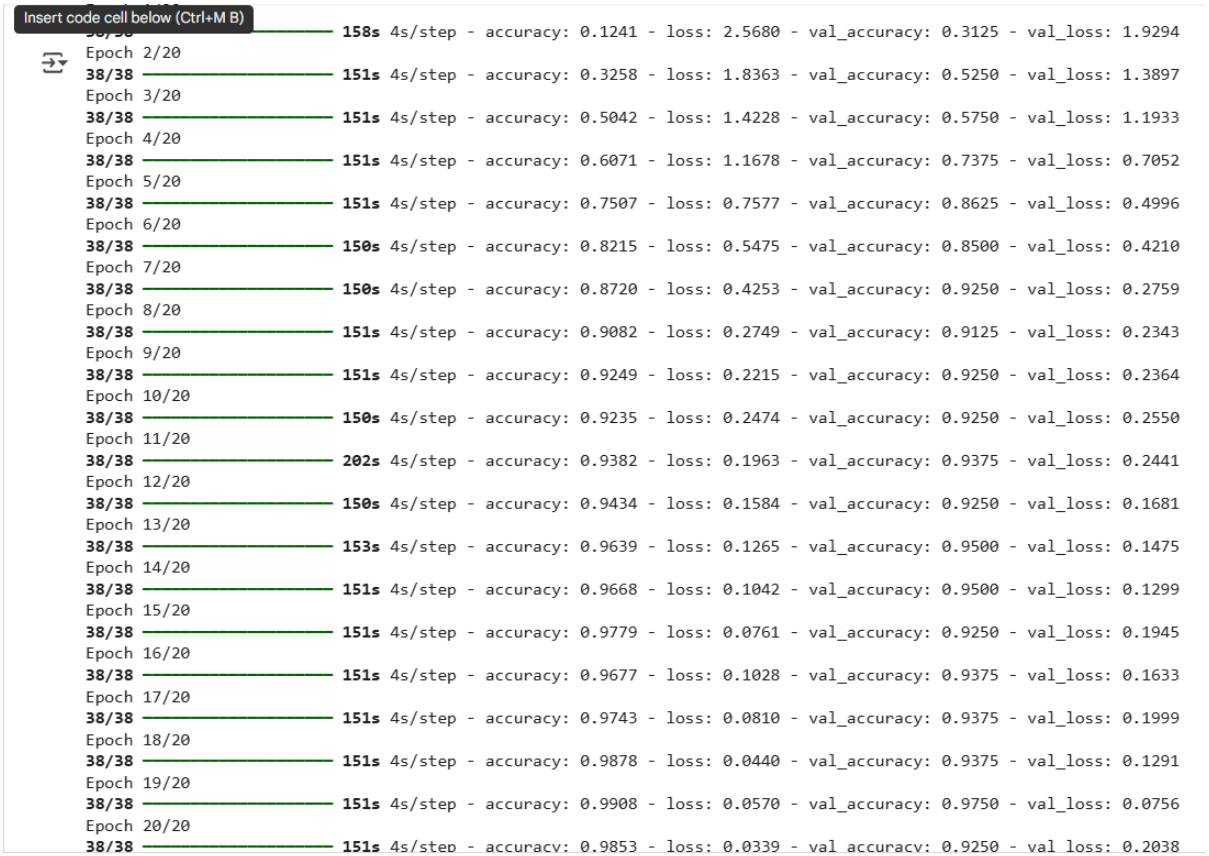

Figure 35: Compiling the Model

Figure 33, Figure 34 and Figure 35 shows the process of training the plant dataset using the CNN with 20 epochs. After compiling the model for 20 epochs we have acquired an accuracy of 0.92%.

Model Testing

Model testing involves evaluating the trained models on unseen test data to assess their performance and generalization ability. This phase typically includes measuring metrics such as accuracy, precision, recall, and F1 score, as well as generating confusion matrices to gain insights into the models' classification performance.

```
import numpy as np
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import load_model
from tensorflow.keras.applications.CNN import preprocess_input

model = load_model('PlantClassification.keras')

def predict_image(img_path, model):
    img = image.load_img(img_path, target_size=(224, 224))

    img_array = image.img_to_array(img)

    img_array = np.expand_dims(img_array, axis=0)

    img_array = preprocess_input(img_array)

    predictions = model.predict(img_array)

    predicted_class = np.argmax(predictions, axis=1)[0]

    return predicted_class

img_path = '/content/drive/MyDrive/Plant Dataset/Test/Ashoka/Ashoka (12).jpg'
predicted_class = predict_image(img_path, model)

print(f"Predicted Class: {predicted_class}")

class_labels = {0: 'Amla', 1: 'Ashoka', 2: 'Ashwagandha',  3: 'Avacado', 4: 'Bamboo', 5: 'Brahmi', 6: 'Neem', 7: 'Nithyapushpa', 8: 'Raktachandini',  9: 'Sapota'}
print(f"Predicted Pose: {class_labels[predicted_class]}")
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 134ms/step
Predicted Class: 1
Predicted Pose: Ashoka
```

Figure 36: Testing the model

Figure 36 shows the snapshot of uploading an image to the model and predicting the output.

RESULTS AND DISCUSSION

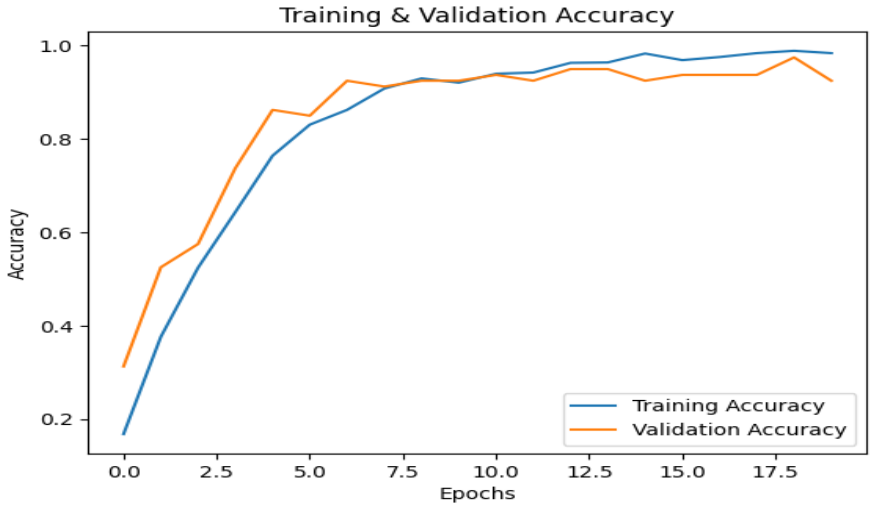

Figure 37: Accuracy Curve

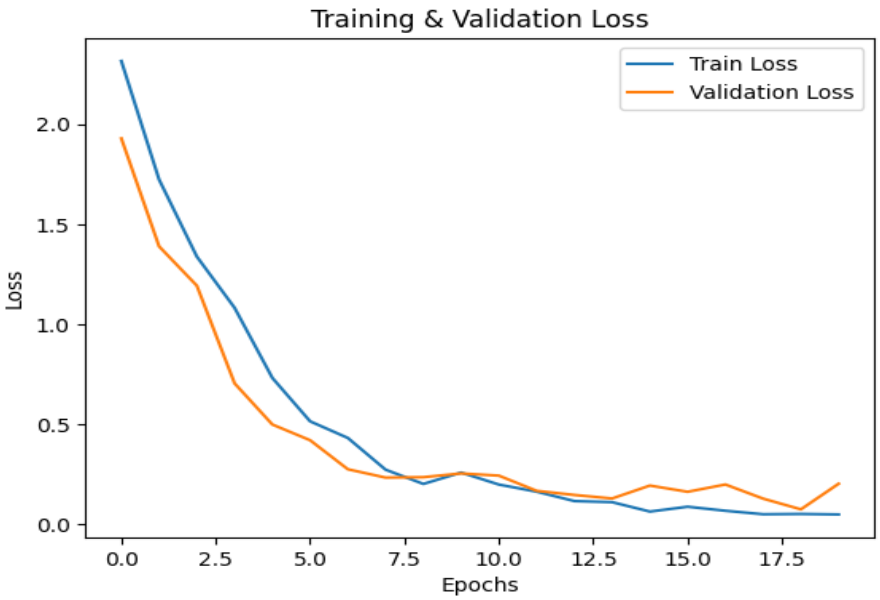

Figure 38: Loss Curve

Figure 37 and Figure 38 shows the accuracy and loss curves, for this model we got an accuracy of 0.92% and a loss of 0.20%.

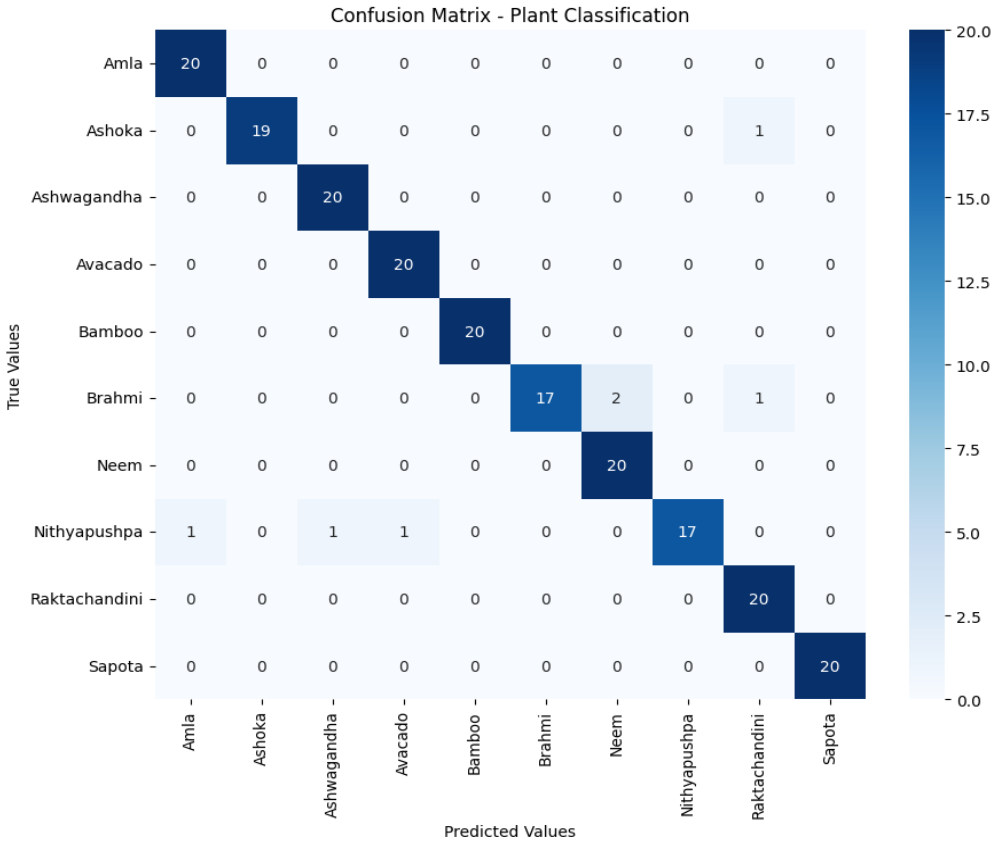

Figure 39: Confusion Matrix

Figure 39 shows the confusion matrix of the model on test data.