Code pour la page accueil:

```
from ._anvil_designer import AcceuilTemplate
from anvil import *
import anvil.server
import anvil.tables as tables
import anvil.tables.query as q
from anvil.tables import app_tables
import anvil.users
from ..compte import compte
from ..annonce import annonce
from ..groupe import groupe



class Acceuil(AcceuilTemplate):
  def __init__(self, **properties):
    anvil.users.login_with_form()
    # Set Form properties and Data Bindings.
    self.init_components(**properties)


  def Gestion_des_comptes_click(self, **event_args):
    self.column_panel_1.clear()
    self.column_panel_1.add_component(compte())

  def creer_des_annonce_click(self, **event_args):
    self.column_panel_1.clear()
    self.column_panel_1.add_component(annonce())


  def Gestion_de_groupe_click(self, **event_args):
```

```
    self.column_panel_1.clear()
    self.column_panel_1.add_component(groupe())
```

code pour compte:

```
from ._anvil_designer import compteTemplate
from anvil import *
import anvil.server
import anvil.tables as tables
import anvil.tables.query as q
from anvil.tables import app_tables
import anvil.users
from ..annonce import annonce




class compte(compteTemplate):
  def __init__(self, **properties):

    self.init_components(**properties)
    self.tableau.set_event_handler('x-refresh-employees', self.refresh_employees)
    self.tableau.items = app_tables.data_1.search()




  def Annonce_click(self, **event_args):
    self.content_panel.clear()
    self.content_panel.add_component(annonce())

  def text_box_1_lost_focus(self, **event_args):
```

```
  self.data_2.rows_per_page = 1 + int(self.text_box_1.text)
  pass

def button_employee_add_click(self, **event_args):
  """This method is called when the button is clicked"""

  # Split the employee name into first and last



  # Add the employee to the database
  anvil.server.call(
    'add_locataire',
    Nom_et_Prenom=self.text_box_2.text,
    No_apartement=self.text_box_5.text,
    Date_entree=self.date_picker_1.date,
    E_mail=self.text_box_4.text,
    Groupe=self.text_box_6.text
  )

  # Refresh the employee data in the Data Grid
  self.tableau.items = anvil.server.call('get_employees')

  # Clear the add row's input components
  self.text_box_2.text = ''
  self.text_box_5.text = ''
  self.date_picker_1.text = ''
  self.text_box_4.text = ''
  self.text_box_6.text = ''

def search(self, **event_args):
  self.tableau.items = anvil.server.call(
```

```
    'search_data_1',
    self.text_box_search.text
  )
  pass


 def refresh_employees(self, **event_args):
  self.tableau.items = anvil.server.call('get_employees')
```

code pour la page annonce:

```
from ._anvil_designer import annonceTemplate
from anvil import *
import anvil.server
import anvil.tables as tables
import anvil.tables.query as q
from anvil.tables import app_tables
import anvil.users
from datetime import datetime, date



class annonce(annonceTemplate):
 def __init__(self, **properties):
  # Set Form properties and Data Bindings.
  self.init_components(**properties)
  self.repeating_panel_2.items = app_tables.sent_email.search()
  # Any code you write here will run before the form opens.
  self.drop_down_1.items =[(r['Groupe'],  r)  for r in app_tables.data_1.search()]



 def button_8_click(self, **event_args):
```

```
    # Recherche des adresses correspondantes
    result = anvil.server.call('search_email', self.drop_down_1.selected_value)
    # Liste pour stocker les adresses déjà traitées
    processed_emails = []
    for row in result:
        email = row["E_mail"]
        # Vérification si l'adresse a déjà été traitée
        if email not in processed_emails:
            # Envoi de l'e-mail
            anvil.server.call('send_a_message', to=email, subject=self.text_box_1.text,
content=self.text_area_1.text)
            # Ajout de l'adresse à la liste des adresses traitées
            processed_emails.append(email)
            # Ajout de l'e-mail à la table
            anvil.server.call(
                'add_email',
                To=email,
                subject=self.text_box_1.text,
                content=self.text_area_1.text,
                Date=date.today()
            )




  def drop_down_1_change(self, **event_args):
    """This method is called when an item is selected"""
    Pass
```

Code pour le serveur module :

```
import anvil.email
import anvil.users
```

```
import anvil.tables as tables
import anvil.tables.query as q
from anvil.tables import app_tables
import anvil.server
from datetime import datetime, date

@anvil.server.callable
def get_employees():
  return app_tables.data_1.search()



@anvil.server.callable
def search_data_1(query):
  result = app_tables.data_1.search()
  if query:
    result = [
      x for x in result
      if query in x['Nom_et_Prenom']
      or query in str(x['No_apartement'])
      or query in x['E_mail']
      or query in x['Groupe']
    ]
  return result


@anvil.server.callable
def add_locataire(Nom_et_Prenom, No_apartement, Date_entree, E_mail, Groupe):
  app_tables.data_1.add_row(Nom_et_Prenom=Nom_et_Prenom, No_apartement=No_apartement, Date_entree=Date_entree, E_mail=E_mail, Groupe=Groupe)


@anvil.server.callable
def edit_data_1(Nom_et_Prenom, No_apartement, Date_entree, E_mail, Groupe):
```

```
  data_1.update(Nom_et_Prenom=Nom_et_Prenom, No_apartement=No_apartement,
Date_entree=Date_entree, E_mail=E_mail, Groupe=Groupe)


@anvil.server.callable
def add_email(To, subject, content, Date):
 app_tables.sent_email.add_row(To=To, subject=subject, content=content, Date= Date.today() )


@anvil.server.callable
def search_email(query) :
  result = app_tables.data_1.search()
  if isinstance(query, str):
    result= [x for x in result
          if query in x['Groupe']
          ]
  return result


@anvil.server.callable
def send_a_message(to, subject, content):
 anvil.email.send(
   to=to,
   subject=subject,
   text=content
 )


@anvil.server.callable
def delete_employee(Nom_et_prenom):
 Nom_et_prenom.delete()
```