Import **necessary modules**

* Import sqlite3 for database operations• Import necessary modules
* Import sqlite3 for database operations
* Import tkinter and messagebox for GUI components
* • Connect to the database
* Establish a connection to the SQLite database (recipes.db)
* Create a cursor object to execute SQL commands
* Create a table named recipes with columns: id, name, ingredients, instructions, category
* Commit changes to the database and close the connection
* • Create the main application window
* Initialize the main window using tkinter
* Set the title of the window
* • Create frames for different sections
* Main menu frame (frame\_main)
* Add Recipe frame (frame\_add)
* Search Recipes frame (frame\_search)
* View All Recipes frame (frame\_view)
* • Function to switch frames
* Define show\_frame(frame) function to display the specified frame and hide others
* • Main menu buttons
* Create buttons for "Add Recipe," "Search Recipes," "View All Recipes," and "Exit"
* Pack these buttons into frame\_main
* Assign each button a command to display the corresponding frame
* • Add Recipe screen elements
* Create labels and entry fields for "Recipe Name," "Ingredients," "Instructions," and "Category"
* Pack these elements into frame\_add
* Define save\_recipe() function to:
* Retrieve user input from entry fields
* Validate input (ensure no fields are empty)
* Connect to the database and insert the recipe into the recipes table
* Commit changes and close the connection
* Show a success message using messagebox
* Clear input fields for new entries
* Show an error message if any fields are empty
* Add "Save" and "Cancel" buttons to frame\_add
* "Save" button calls save\_recipe()
* "Cancel" button switches back to frame\_main
* • Search Recipes screen elements
* Create a label and entry field for search queries
* Create a Listbox to display search results
* Pack these elements into frame\_search
* Define search\_recipes() function to:
* Retrieve user input from the search entry field
* Connect to the database and search for recipes matching the query (by name, ingredients, or category)
* Populate the Listbox with search results
* Bind Listbox items to display recipe details on double-click
* Define show\_recipe\_details(recipe\_id) function to:
* Connect to the database and fetch details of the selected recipe by recipe\_id
* Display recipe details using messagebox
* Add "Search" and "Back" buttons to frame\_search
* "Search" button calls search\_recipes()
* "Back" button switches back to frame\_main
* • View All Recipes screen elements
* Create a Listbox to display all recipes
* Pack the Listbox into frame\_view
* Define view\_all\_recipes() function to:
* Connect to the database and fetch all recipes
* Populate the Listbox with all recipes
* Bind Listbox items to display recipe details on double-click (using show\_recipe\_details(recipe\_id))
* Add "Refresh" and "Back" buttons to frame\_view
* "Refresh" button calls view\_all\_recipes()
* "Back" button switches back to frame\_main
* • Run the main application loop
* Start the GUI event loop using root.mainloop()
* Import tkinter and messagebox for GUI components

Connect **to the database**

* Establish a connection to the SQLite database (recipes.db)
* Create a cursor object to execute SQL commands
* Create a table named recipes with columns: id, name, ingredients, instructions, category
* Commit changes to the database and close the connection

Create **the main application window**

* Initialize the main window using tkinter
* Set the title of the window

Create **frames for different sections**

* Main menu frame (frame\_main)
* Add Recipe frame (frame\_add)
* Search Recipes frame (frame\_search)
* View All Recipes frame (frame\_view)

Function**to switch frames**

* Define show\_frame(frame) function to display the specified frame and hide others

Main **menu buttons**

* Create buttons for "Add Recipe," "Search Recipes," "View All Recipes," and "Exit"
* Pack these buttons into frame\_main
* Assign each button a command to display the corresponding frame

Add **Recipe screen elements**

* Create labels and entry fields for "Recipe Name," "Ingredients," "Instructions," and "Category"
* Pack these elements into frame\_add
* Define save\_recipe() function to:
  + Retrieve user input from entry fields
  + Validate input (ensure no fields are empty)
  + Connect to the database and insert the recipe into the recipes table
  + Commit changes and close the connection
  + Show a success message using messagebox
  + Clear input fields for new entries
  + Show an error message if any fields are empty
* Add "Save" and "Cancel" buttons to frame\_add
  + "Save" button calls save\_recipe()
  + "Cancel" button switches back to frame\_main

Search **Recipes screen elements**

* Create a label and entry field for search queries
* Create a Listbox to display search results
* Pack these elements into frame\_search
* Define search\_recipes() function to:
  + Retrieve user input from the search entry field
  + Connect to the database and search for recipes matching the query (by name, ingredients, or category)
  + Populate the Listbox with search results
* Bind Listbox items to display recipe details on double-click
* Define show\_recipe\_details(recipe\_id) function to:
  + Connect to the database and fetch details of the selected recipe by recipe\_id
  + Display recipe details using messagebox
* Add "Search" and "Back" buttons to frame\_search
  + "Search" button calls search\_recipes()
  + "Back" button switches back to frame\_main

View **All Recipes screen elements**

* Create a Listbox to display all recipes
* Pack the Listbox into frame\_view
* Define view\_all\_recipes() function to:
  + Connect to the database and fetch all recipes
  + Populate the Listbox with all recipes
* Bind Listbox items to display recipe details on double-click (using show\_recipe\_details(recipe\_id))
* Add "Refresh" and "Back" buttons to frame\_view
  + "Refresh" button calls view\_all\_recipes()
  + "Back" button switches back to frame\_main

Run **the main application loop**

* Start the GUI event loop using root.mainloop()