# EXERCISE 8: TABLES AND CHARTS

**Create a Flet app to add daily expenses in 4 different categories and display each entries in a table. Update a pie chart that displays the share of expenses in the different categories.**

import flet as ft

def main(page: ft.Page):

    page.title = "Charts and Tables"

    def add\_data(e):

        table.rows.append(

            ft.DataRow(

                cells=[

                    ft.DataCell(ft.Text(list\_categories[categories.selected\_index])),

                    ft.DataCell(ft.Text(amount.value)),

                ],))

        colors = [ft.colors.AMBER\_200, ft.colors.TEAL\_ACCENT\_200, ft.colors.GREEN\_ACCENT\_200, ft.colors.DEEP\_ORANGE\_200]

        chart.sections.append(

            ft.PieChartSection(

               value=amount.value,

               title=list\_categories[categories.selected\_index],

               title\_position=2.0,

               color=colors[categories.selected\_index]

            ))

        page.update()

    list\_categories=["Clothes","Entertainment","Food","Transport"]

    # Create categories for the chart

    categories = ft.AutoComplete(

        suggestions=[

            ft.AutoCompleteSuggestion(value="Clothes"),

            ft.AutoCompleteSuggestion(value="Entertainment"),

            ft.AutoCompleteSuggestion(value="Food"),

            ft.AutoCompleteSuggestion(value="Transport"),

        ],

        on\_select=lambda e: print(e.control.selected\_index, e.selection),

    )

    amount=ft.TextField(label="Amount: ", width=200)

    add\_btn = ft.ElevatedButton("Add", on\_click=add\_data)

    table = ft.DataTable(

        columns=[

            # Identifier for each row

            ft.DataColumn(ft.Text("Category")),

            ft.DataColumn(ft.Text("Amount"))

        ],

        rows=[]

    )

    chart = ft.PieChart(

        width=300,

        height=300,

        sections=[],    # Data will be added later

        sections\_space=1,

        center\_space\_radius=0,

        expand=True,

    )

    page.add(

        ft.Row([ft.Column([categories],width=200),amount,add\_btn]),

        ft.Row([chart, table]),

    )

    page.update()

ft.app(main)

The result of executing the code and adding the values shown in the left-sided table is the following:

A screenshot of a computer

Description automatically generated