

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:

