# EXERCISE 6: ASYNC COUNTDOWN TIMER

import flet as ft

import asyncio

*# Define a new class Countdown where we define the timer*

class Countdown(ft.Column):

    def \_\_init\_\_(self, seconds):

        super().\_\_init\_\_()

        self.initial\_seconds = seconds

        self.seconds = seconds

        self.running = False

*# Display elements for the countdown timer*

        self.time\_display = ft.TextField(f"Initial value: {self.format\_time(self.initial\_seconds)}", read\_only=True, width=200)

*# Control buttons*

        self.start\_btn = ft.ElevatedButton("Start", on\_click=self.start)

        self.pause\_btn = ft.ElevatedButton("Pause", on\_click=self.pause)

        self.reset\_btn = ft.ElevatedButton("Reset", on\_click=self.reset)

*# Define the progress bar*

        self.progress = ft.ProgressBar(value=0.0)

        self.controls = [

            ft.Row([self.time\_display], alignment=ft.MainAxisAlignment.CENTER),

            ft.Row([self.start\_btn, self.pause\_btn, self.reset\_btn], alignment=ft.MainAxisAlignment.CENTER),

            self.progress

        ]

    def start(self, e=None):

        self.running = True

        self.page.run\_task(self.update\_timer)

    def pause(self, e=None):

        self.running = False

    def reset(self, e=None):

        self.seconds = self.initial\_seconds

        self.running = False

        self.update\_display()

def update\_display(self):

        self.time\_display.value = self.format\_time(self.seconds)

        self.progress.value = (self.initial\_seconds - self.seconds) / self.initial\_seconds

        self.update()

    def format\_time(self,seconds):

        mins, secs = divmod(seconds, 60)

        return "{:02d}:{:02d}".format(mins, secs)

    async def update\_timer(self):

        while self.seconds > 0 and self.running:

            self.update\_display()

            await asyncio.sleep(1)

            self.seconds -= 1

            if not self.running:

                break

            self.update\_display()

*# Define the main function*

def main(page: ft.Page):

    page.title = "Async Countdown"

    page.update()

*# Create three tabs, with a timer in each*

    tabs = ft.Tabs(

        selected\_index=0,

        animation\_duration=300,

        tabs = [

            ft.Tab(text="Timer 1", content = Countdown(120)),

            ft.Tab(text="Timer 2", content = Countdown(140)),

            ft.Tab(text="Timer 3", content = Countdown(160))

        ]

    )

    page.add(tabs)

A screenshot of a computer

Description automatically generatedft.app(main)