Loan calculator

Code:

app.py 4

C: > loan_calculator_app > app.py > ...

```python
import streamlit as st
import pandas as pd
import numpy as np
import altair as alt

st.set_page_config(page_title="Loan Calculator", page_icon="💸", layout="wide")

# --- Inputs ---
st.sidebar.header("Loan Inputs")
name = st.sidebar.text_input("Name", "Kavana")
age = st.sidebar.number_input("Age", 18, 100, 25)
loan_amount = st.sidebar.number_input("Loan Amount", min_value=10000, value=500000, step=10000)
deposit = st.sidebar.slider("Deposit", 0, loan_amount, 50000)
interest = st.sidebar.number_input("Interest Rate (%)", 1.0, 20.0, 7.5, step=0.1)
years = st.sidebar.slider("Duration (Years)", 1, 30, 10)
extra_payment = st.sidebar.number_input("Extra Payment per Month", 0, 50000, 0, step=500)
show_schedule = st.sidebar.checkbox("Show Amortization Schedule", True)

# --- Calculations ---
principal = loan_amount - deposit
months = years * 12
monthly_rate = interest / 100 / 12
emi = (principal * monthly_rate * (1 + monthly_rate) ** months) / ((1 + monthly_rate) ** months - 1)

balance = principal
schedule = []
for m in range(1, months + 1):
    interest_component = balance * monthly_rate
    principal_component = emi - interest_component + extra_payment
    balance -= principal_component
    if balance < 0:
        balance = 0
    schedule.append([m, emi + extra_payment, interest_component, principal_component, balance])
    if balance <= 0:
        break
```

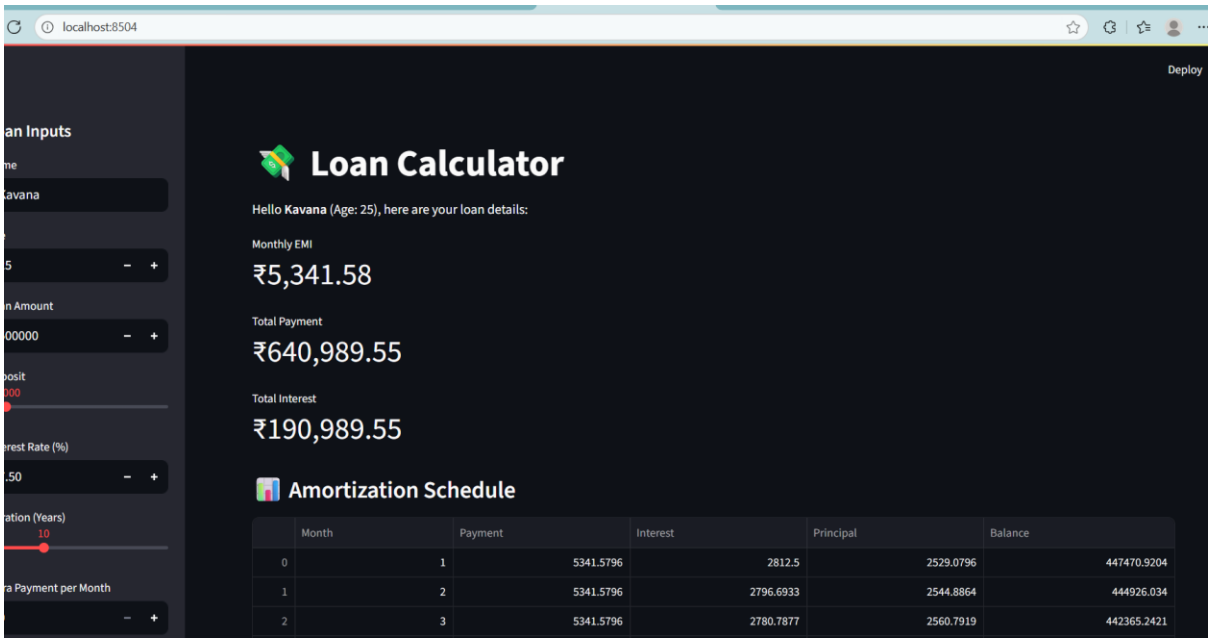
app.py 4

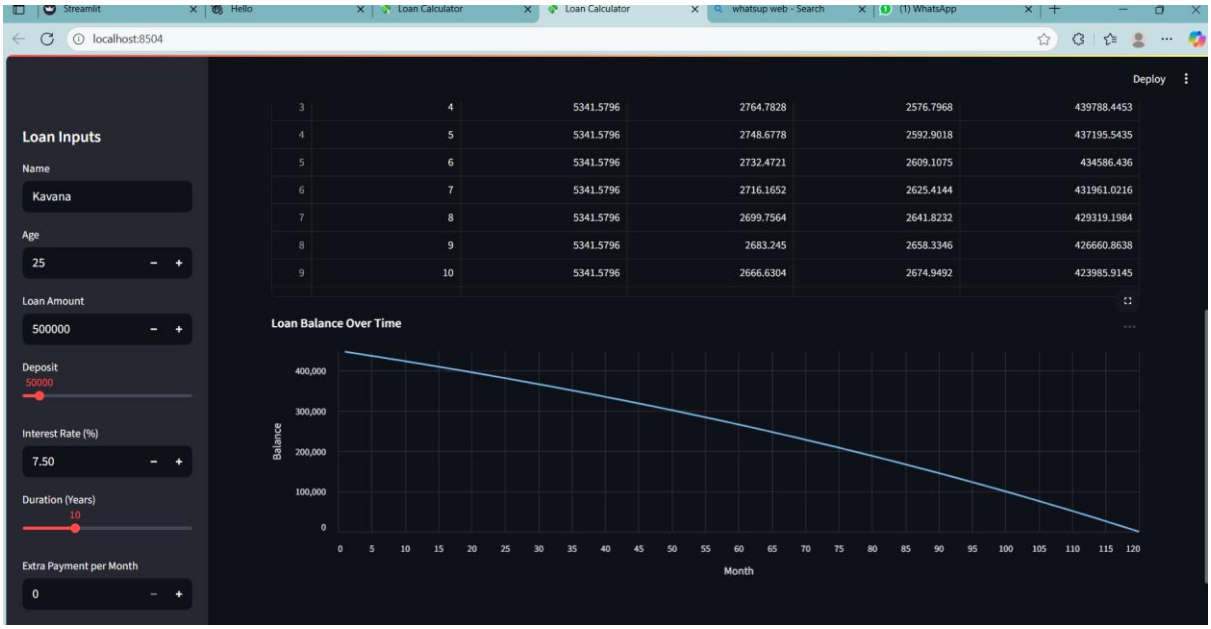
C: > loan_calculator_app > app.py > ...

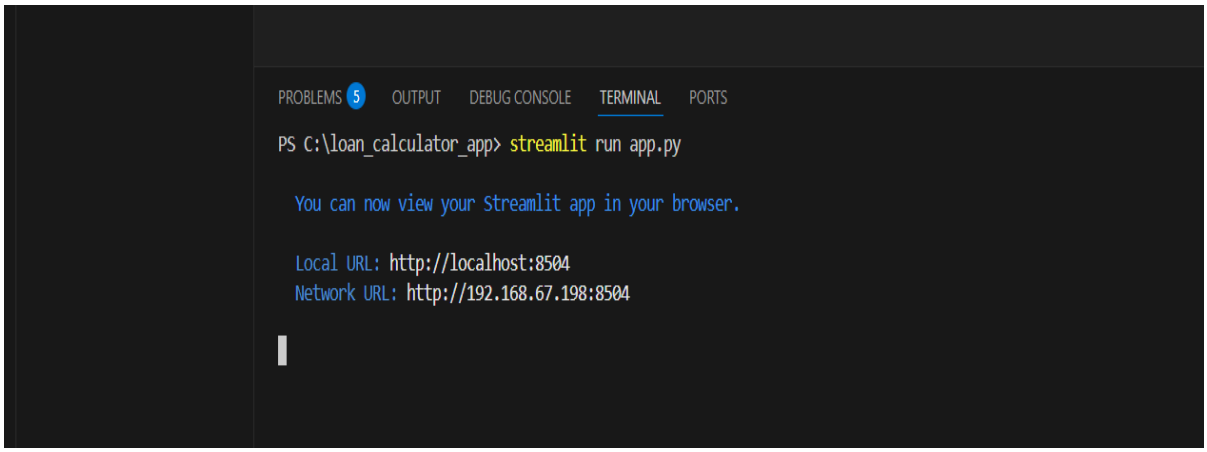
```python
df = pd.DataFrame(schedule, columns=["Month", "Payment", "Interest", "Principal", "Balance"])

# --- Output ---
st.title("💸 Loan Calculator")
st.write(f"Hello **{name}** (Age: {age}), here are your loan details:")

st.metric("Monthly EMI", f"₹{emi:,.2f}")
st.metric("Total Payment", f"₹{df['Payment'].sum():,.2f}")
st.metric("Total Interest", f"₹{df['Interest'].sum():,.2f}")

if show_schedule:
    st.subheader("📊 Amortization Schedule")
    st.dataframe(df, use_container_width=True)

    chart = alt.Chart(df).mark_line().encode(
        x="Month",
        y="Balance"
    ).properties(title="Loan Balance Over Time")
    st.altair_chart(chart, use_container_width=True)
```

Loan calculator

Output:







Loan calculator