

```
import streamlit as st
from calculator import Calculator
import math

# Initialize session state
if "calculator" not in st.session_state:
    st.session_state.calculator = Calculator()

calc = st.session_state.calculator

# Page configuration
st.set_page_config(
    page_title="Calculator App",
    page_icon="🧮",
    layout="wide"
)

# Custom CSS for better styling
st.markdown("""
<style>
.calculator-container {
    background-color: #f0f2f6;
    padding: 20px;
    border-radius: 10px;
    margin: 10px 0;
}
.button-container {
    display: flex;
    gap: 5px;
    margin: 5px 0;
}
</style>
""")
```

```
.calc-button {  
    background-color: #4CAF50;  
    color: white;  
    padding: 10px 15px;  
    border: none;  
    border-radius: 5px;  
    cursor: pointer;  
    font-size: 16px;  
    margin: 2px;  
}  
.operator-button {  
    background-color: #FF9800;  
}  
.function-button {  
    background-color: #2196F3;  
}  
.clear-button {  
    background-color: #f44336;  
}  
.equals-button {  
    background-color: #4CAF50;  
}  
</style>  
"""", unsafe_allow_html=True)
```

```
# Main title  
st.title("🧮 Advanced Calculator")  
  
# Create two columns  
col1, col2 = st.columns([2, 1])
```

```
with col1:
    st.subheader("Calculator")

    # Display current expression and result
    st.markdown("### Expression")
    expression_display = calc.current_expression if calc.current_expression else "0"
    st.text_input("", value=expression_display, key="expression_display", disabled=True)

    st.markdown("### Result")
    result_display = str(calc.result) if calc.result != 0 else "0"
    st.text_input("", value=result_display, key="result_display", disabled=True)

    # Calculator buttons
    st.markdown("### Number Pad")

    # Row 1: Function buttons
    col1, col2, col3, col4 = st.columns(4)

    with col1:
        if st.button("sin", key="sin"):
            calc.add_to_expression("sin(")

    with col2:
        if st.button("cos", key="cos"):
            calc.add_to_expression("cos(")

    with col3:
        if st.button("tan", key="tan"):
            calc.add_to_expression("tan(")

    with col4:
        if st.button("√", key="sqrt"):
            calc.add_to_expression("sqrt(")

    # Row 2: More function buttons
```

```
col1, col2, col3, col4 = st.columns(4)

with col1:
    if st.button("ln", key="ln"):
        calc.add_to_expression("ln(")

with col2:
    if st.button("log", key="log"):
        calc.add_to_expression("log(")

with col3:
    if st.button("^", key="power"):
        calc.add_to_expression("^")

with col4:
    if st.button("π", key="pi"):
        calc.add_to_expression(str(math.pi))
```

Row 3: Numbers 7-9 and divide

```
col1, col2, col3, col4 = st.columns(4)

with col1:
    if st.button("7", key="7"):
        calc.add_to_expression("7")

with col2:
    if st.button("8", key="8"):
        calc.add_to_expression("8")

with col3:
    if st.button("9", key="9"):
        calc.add_to_expression("9")

with col4:
    if st.button("÷", key="divide"):
        calc.add_to_expression("÷")
```

Row 4: Numbers 4-6 and multiply

```
col1, col2, col3, col4 = st.columns(4)
```

```
with col1:  
    if st.button("4", key="4"):  
        calc.add_to_expression("4")  
  
with col2:  
    if st.button("5", key="5"):  
        calc.add_to_expression("5")  
  
with col3:  
    if st.button("6", key="6"):  
        calc.add_to_expression("6")  
  
with col4:  
    if st.button("x", key="multiply"):  
        calc.add_to_expression("x")  
  
  
# Row 5: Numbers 1-3 and subtract  
col1, col2, col3, col4 = st.columns(4)  
  
with col1:  
    if st.button("1", key="1"):  
        calc.add_to_expression("1")  
  
with col2:  
    if st.button("2", key="2"):  
        calc.add_to_expression("2")  
  
with col3:  
    if st.button("3", key="3"):  
        calc.add_to_expression("3")  
  
with col4:  
    if st.button("-", key="subtract"):  
        calc.add_to_expression("-")  
  
  
# Row 6: 0, decimal, equals, and add  
col1, col2, col3, col4 = st.columns(4)  
  
with col1:
```

```
if st.button("0", key="0"):  
    calc.add_to_expression("0")  
  
with col2:  
  
    if st.button(".", key="decimal"):  
        calc.add_to_expression(".")  
  
with col3:  
  
    if st.button("=", key="equals"):  
        calc.calculate()  
  
with col4:  
  
    if st.button("+", key="add"):  
        calc.add_to_expression("+")
```

Row 7: Clear and delete

```
col1, col2, col3, col4 = st.columns(4)  
  
with col1:  
  
    if st.button("C", key="clear"):  
        calc.clear()  
  
with col2:  
  
    if st.button("⌫", key="delete"):  
        calc.delete_last()  
  
with col3:  
  
    if st.button("(", key="open_paren"):  
        calc.add_to_expression("(")  
  
with col4:  
  
    if st.button(")", key="close_paren"):  
        calc.add_to_expression(")")
```

with col2:
st.subheader("Calculation History")

Display history

```
if calc.history:
    for i, entry in enumerate(reversed(calc.history[-10:])):
        st.write(f"**{entry['expression']} = {entry['result']}**")
        st.divider()
else:
    st.write("No calculations yet")

# Clear history button
if st.button("Clear History"):
    calc.clear_history()
    st.rerun()

# Instructions
with st.expander("How to Use"):
    st.write(""""
**Basic Operations:**
- Use the number pad to input numbers
- Use +, -, ×, ÷ for basic arithmetic
- Press = to calculate the result

**Scientific Functions:**
- sin, cos, tan: Trigonometric functions (in degrees)
- √: Square root
- ln: Natural logarithm
- log: Common logarithm (base 10)
- ^: Power function
- π: Pi constant

**Other Features:**
- C: Clear current expression
- ✖: Delete last character
```

- (): Parentheses for grouping
 - History: View previous calculations
- """)

```
# Auto-refresh to update display
st.rerun()
```