

```
import streamlit as st
import pandas as pd
```

```
def calculate_euc_job_cost(base_hours, complexity, service_type, additional_factors):
    cost_per_hour = 100
    complexity_multipliers = {"Basic": 1.0, "Standard": 1.2,
                              "Advanced": 1.5, "Extreme": 2.0}
    service_type_multipliers = {"Remote Support": 0.8,
                                "Onsite Standard": 1.0, "Onsite Urgent": 1.5,
                                "Multi-Site Deployment": 1.3}
    cost_factors = {"Stairs or No Lift": 1.1, "Security Checkpoints": 1.15,
                    "Remote Area": 1.3, "Custom Desk Configurations": 1.2,
                    "Heavy Cable Management": 1.25, "Data Migration (Basic)":
                    1.1, "Data Migration (Full)": 1.5, "Government Clearance Delays": 1.4,
                    "Disaster Recovery Setup": 1.6}
```

```
    complexity_multiplier = complexity_multipliers.get(complexity, 1.0)
    service_type_multiplier = service_type_multipliers.get(service_type, 1.0)
    additional_multiplier = 1.0
```

```
    for factor in additional_factors:
        additional_multiplier *= cost_factors.get(factor, 1.0)
```

```
    adjusted_hours = base_hours * complexity_multiplier * service_type_multiplier *
        additional_multiplier
    adjusted_hours = round(adjusted_hours)
    total_cost = adjusted_hours * cost_per_hour
```

```
    return adjusted_hours, total_cost
```

```
def main():
    st.title("EUC Job Pricing Calculator")
```

```
    base_hours = st.number_input("Base Estimated Hours", min_value=1, value=10)
    complexity = st.selectbox("Complexity Level", ["Basic", "Standard", "Advanced",
                                                    "Extreme"])
    service_type = st.selectbox("Service Type", ["Remote Support", "Onsite Standard",
                                                  "Onsite Urgent", "Multi-Site Deployment"])
    additional_factors = st.multiselect("Additional Factors", ["Stairs or No Lift", "Security
                                                                Checkpoints", "Remote Area",
                                                                "Custom Desk Configurations",
                                                                "Heavy Cable Management",
                                                                "Data Migration (Basic)",
                                                                "Data Migration (Full)",
                                                                "Government Clearance Delays",
                                                                "Disaster Recovery Setup"])
```

```
    if st.button("Calculate Quote"):
        final_hours, final_cost = calculate_euc_job_cost(base_hours, complexity,
```

```
service_type, additional_factors)
    st.subheader(f"Final Billable Hours: {final_hours} Hours")
    st.subheader(f"Total Cost: AUD ${final_cost:.2f}")
```

```
if name == "main": main()
```