In [23]:

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
import pandas as pd
from sklearn.tree import plot_tree
import matplotlib.pyplot as plt
```

In [24]:

```
# Load pre-processed data
train_df = pd.read_csv("../data/titanic_preprocessed.csv")
X = train_df.drop("Survived", axis=1)
y = train_df["Survived"]
```

In [25]:

```
# Split data into training and validation
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2, rando
print(f"Training set size: {len(X_train)}")
print(f"Validation set size: {len(X_val)}")
```

```
Training set size: 712
Validation set size: 179
```

Question 1: Implement, train, and plot decision tree model

In [26]:

```
# Initialize and train decision tree
model = DecisionTreeClassifier(
    random_state=42,
    max_depth=5,
    min_samples_split=10,
    min_samples_leaf=5
)
model.fit(X_train, y_train)
```

Out[26]:

```
▾ DecisionTreeClassifier
▶ Parameters
```

In [27]:

```
# Predicit on validation set
y_pred = model.predict(X_val)

# Evaluate accuracy
accuracy = accuracy_score(y_val, y_pred)
print(f"Validation Accuracy: {accuracy:.4f}")
```

```
Validation Accuracy: 0.7542
```

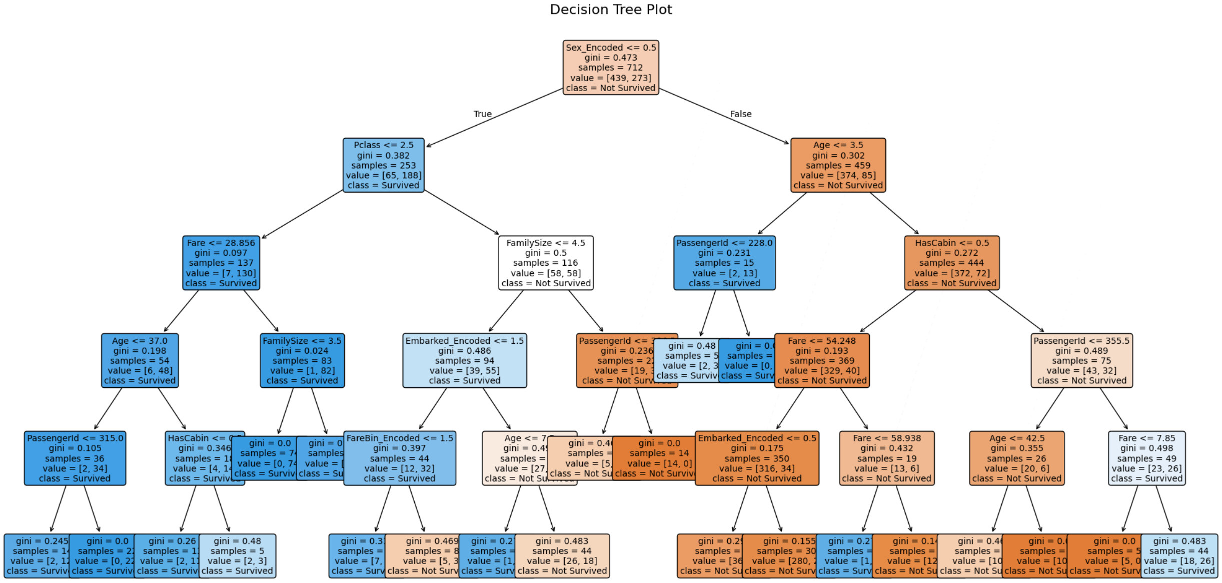
In [28]:

```
# Plot tree

plt.figure(figsize=(20, 10))

plot_tree(model,
          feature_names=X_train.columns,
          class_names=['Not Survived', 'Survived'],
          filled=True,
          rounded=True,
          fontsize=10)

plt.title("Decision Tree Plot", fontsize=16)
plt.tight_layout()
plt.show()
```


Decision Tree Plot

Question 2: Apply 5 fold cross validation to decision tree

In [29]:

```
from sklearn.model_selection import cross_val_score
import numpy as np
```

In [30]:

```
# Combine training and validation sets for cross validation
X_full = pd.concat([X_train, X_val])
y_full = pd.concat([y_train, y_val])
```

In [ ]:

```
# Apply 5 fold cross validation
cv_scores = cross_val_score(
    model,
    X_full,
    y_full,
    cv=5,
    scoring='accuracy'
)
```

In [37]:

```
print("Cross-Validation Scores:", cv_scores)
print(f"Average Accuracy: {cv_scores.mean():.4f}")
```

```
Cross-Validation Scores: [0.82122905 0.78651685 0.86516854 0.79213483 0.7584
2697]
Average Accuracy: 0.8047
```

Question 3: Implement 5 fold cross validation on Random Forest model

In [33]:

```python
from sklearn.ensemble import RandomForestClassifier
```

In [34]:

```python
# Initialize Random Forest model
forest_model = RandomForestClassifier(
    n_estimators=100,
    random_state=42,
    max_depth=5,
    min_samples_split=10,
    min_samples_leaf=5
)
```

In [35]:

```python
# Apply 5 fold cross validation
forest_cv_scores = cross_val_score(
    forest_model,
    X_full,
    y_full,
    cv=5,
    scoring='accuracy'
)
```

In [38]:

```python
print("Random Forest Cross-Validation Scores:", forest_cv_scores)
print(f"Average Accuracy: {forest_cv_scores.mean():.4f}")
```

```
Random Forest Cross-Validation Scores: [0.81005587 0.80898876 0.84269663 0.8
258427  0.83146067]
Average Accuracy: 0.8238
```

In [ ]:

```python

```