```
import streamlit as st
import numpy as np
import joblib

# Load pre-trained models
#@st.cache(allow_output_mutation=True)
def load_model(model_path):
    return joblib.load(model_path)

models = {
    'KNN': load_model('knn_model.pkl'),
    'Decision Tree': load_model('dt_model.pkl'),
    'SVM': load_model('svm_model.pkl'),
    'Logistic Regression': load_model('lr_model.pkl'),
    'Naive Bayes': load_model('nb_model.pkl'),
    'Random Forest': load_model('rf_model.pkl')
}

# Title of the web app
st.title("Loan Prediction Web Application")

# Create input fields for the form (example input fields based on loan
data)
st.write("Please fill out the form to predict loan approval")

# Example form fields - modify these based on your dataset features
Gender = st.selectbox("Gender", ("Male", "Female"))
Married = st.selectbox("Married", ("Yes", "No"))
Dependents = st.selectbox("Dependents", ("0", "1", "2", "3+"))
Education = st.selectbox("Education", ("Graduate", "Not Graduate"))
Self_Employed = st.selectbox("Self Employed", ("Yes", "No"))
ApplicantIncome = st.number_input("Applicant Income", min_value=0)
CoapplicantIncome = st.number_input("Coapplicant Income", min_value=0)
LoanAmount = st.number_input("Loan Amount", min_value=0)
Loan_Amount_Term = st.number_input("Loan Amount Term", min_value=0)
Credit_History = st.selectbox("Credit History", ("0", "1"))
Property_Area = st.selectbox("Property Area", ("Urban", "Semiurban",
"Rural"))

# Create a dictionary to map inputs for model use
form_data = {
    "Gender": 1 if Gender == "Male" else 0,
    "Married": 1 if Married == "Yes" else 0,
    "Dependents": 0 if Dependents == "0" else (1 if Dependents == "1"
else (2 if Dependents == "2" else 3)),
    "Education": 0 if Education == "Graduate" else 1,
    "Self_Employed": 1 if Self_Employed == "Yes" else 0,
    "ApplicantIncome": ApplicantIncome,
    "CoapplicantIncome": CoapplicantIncome,
    "LoanAmount": LoanAmount,
    "Loan_Amount_Term": Loan_Amount_Term,
    "Credit_History": int(Credit_History),
    "Property_Area": 0 if Property_Area == "Urban" else (1 if
Property_Area == "Semiurban" else 2)
}

# Convert input data to a DataFrame-like format for prediction
input_data = np.array([list(form_data.values())]).reshape(1, -1)
```

```
# Prediction logic: when user submits form
if st.button("Predict Loan Status"):
    st.write("Predictions from the models:")
    for model_name, model in models.items():
        prediction = model.predict(input_data)
        prediction_result = "Approved" if prediction[0] == 1 else "Not 
Approved"
        st.write(f"{model_name}: {prediction_result}")
```