EXP NO: 9	MINI PROJECT - BMI PREDICTION WEB APPLICATION USING RANDOM FOREST AND SUPPORT VECTOR MACHINE (SVM)

AIM:

To develop a **BMI (Body Mass Index) Prediction Web Application** using **Machine Learning algorithms** such as **Random Forest Regressor** and **Support Vector Regressor (SVM)**, implemented with **Streamlit**, that predicts a person's BMI based on their **age, height, and weight**, and classifies them into respective BMI categories.

.

ALGORITHM:

1. Import necessary Python libraries like pandas, numpy, streamlit, and sklearn.
2. Create a **synthetic dataset** with random values for **Age**, **Height**, and **Weight**.
3. Calculate **BMI** using the formula:

$$BMI = \frac{Weight}{(Height/100)^2}$$

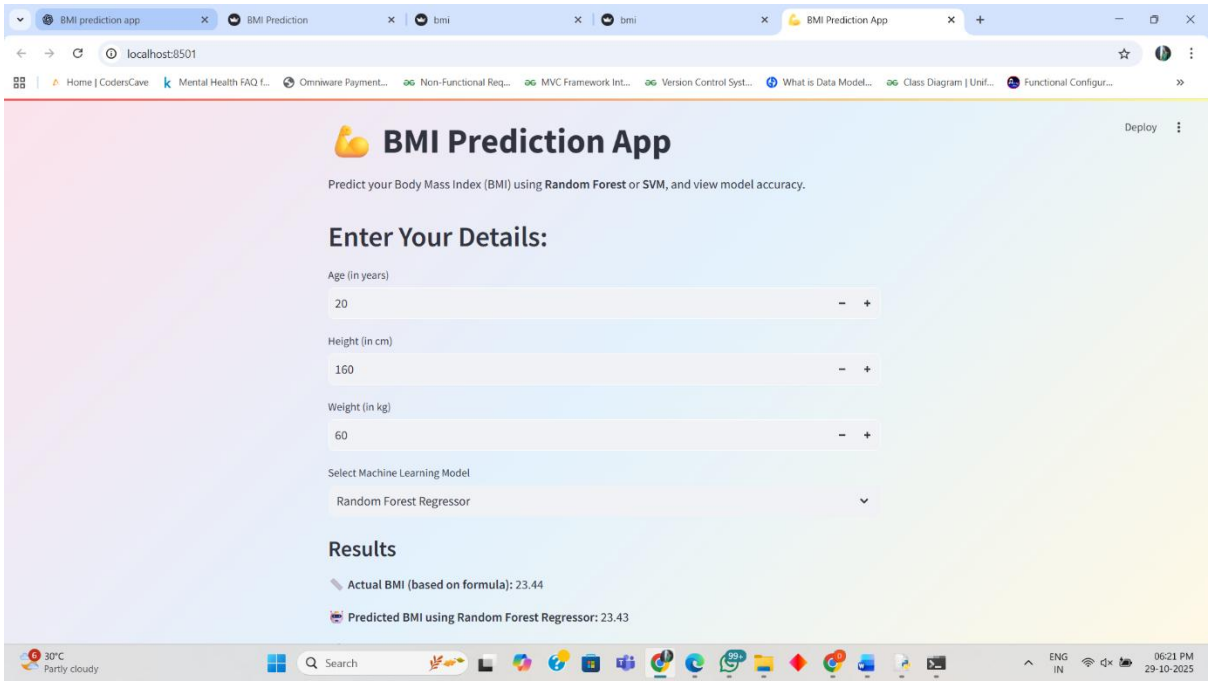
4. Define **features (X)** as Age, Height, and Weight, and **target (y)** as BMI.
5. Use **Random Forest Regressor** and **Support Vector Regressor (SVM)** models from sklearn.
6. Apply **StandardScaler** for SVM to normalize input features.
7. Train both models using the prepared dataset.
8. Take **user inputs** (Age, Height, Weight) from the Streamlit interface.
9. Predict BMI using the selected machine learning model.
10. Calculate **accuracy** using the **R^2 score** to evaluate model performance.
11. Display **Actual BMI**, **Predicted BMI**, **Model Accuracy**, and **BMI Category** on the web app.
12. Customize the interface with a **pastel gradient background** for a modern and appealing look.

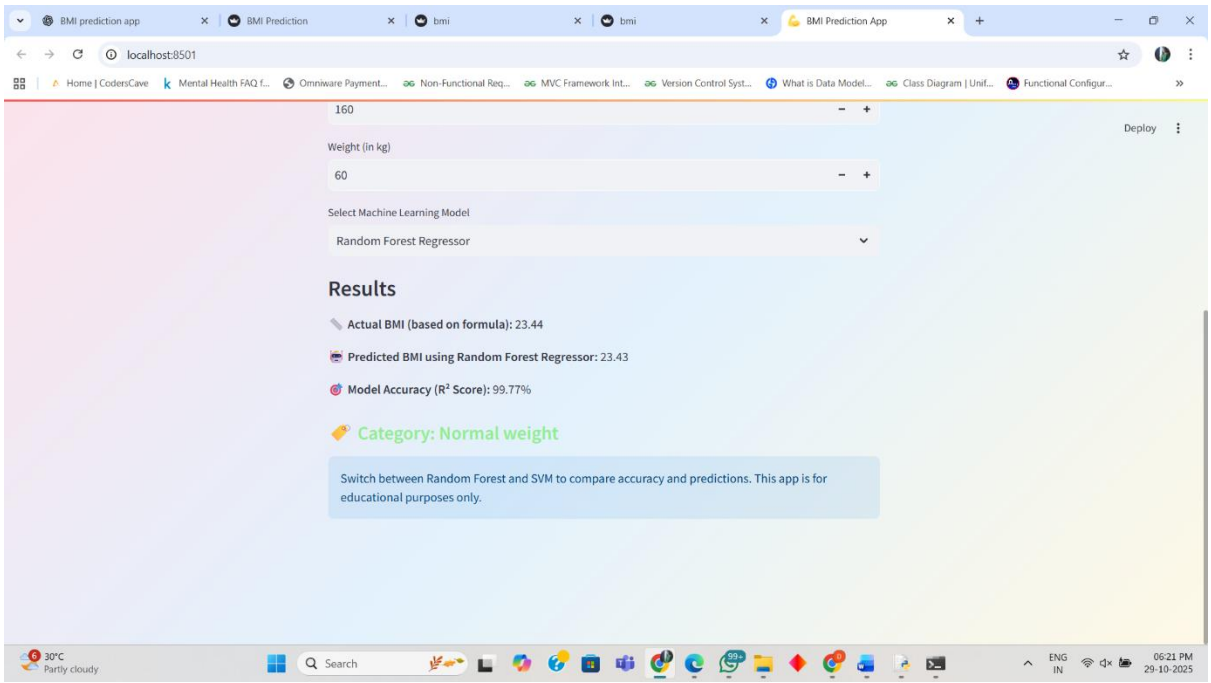
PROGRAM:

```
import streamlit as st
import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestRegressor
from sklearn.svm import SVR
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import r2_score
```

```
# 🌸 Page Configuration
st.set_page_config(page_title="BMI Prediction App", page_icon="💪", layout="centered")

# 🌈 Custom pastel background
page_bg = """
<style>
[data-testid="stAppViewContainer"] {
   background: linear-gradient(135deg, #fce3ec, #e3f2fd, #fff9e6);
   background-attachment: fixed;
}
[data-testid="stHeader"] {
   background: rgba(0, 0, 0, 0);
}
[data-testid="stSidebar"] {
   background-color: #f5f5f5;
}
h1, h2, h3, h4 {
   color: #444444;
   font-family: 'Poppins', sans-serif;
}
</style>
"""
st.markdown(page_bg, unsafe_allow_html=True)

# 🌟 Title
st.title("💪 BMI Prediction App")
st.write("Predict your Body Mass Index (BMI) using **Random Forest** or **SVM**, and view
model accuracy.")

# 🧍 User Inputs
st.header("Enter Your Details:")
age = st.number_input("Age (in years)", min_value=1, max_value=100, value=20)
height = st.number_input("Height (in cm)", min_value=100, max_value=250, value=160)
weight = st.number_input("Weight (in kg)", min_value=20, max_value=200, value=60)

# 💡 Actual BMI Calculation
bmi = weight / ((height / 100) ** 2)

# 📊 Synthetic Training Data
np.random.seed(42)
data = {
   "Age": np.random.randint(18, 60, 200),
   "Height": np.random.randint(140, 190, 200),
   "Weight": np.random.randint(40, 100, 200)
}
df = pd.DataFrame(data)
df["BMI"] = df["Weight"] / ((df["Height"] / 100) ** 2)

X = df[["Age", "Height", "Weight"]]
```

```
y = df["BMI"]

model_choice = st.selectbox(
    "Select Machine Learning Model",
    ("Random Forest Regressor", "Support Vector Regressor (SVM)")
)

if model_choice == "Random Forest Regressor":
    model = RandomForestRegressor(n_estimators=100, random_state=42)
    model.fit(X, y)
    y_pred = model.predict(X)
    predicted_bmi = model.predict([[age, height, weight]])[0]
    accuracy = r2_score(y, y_pred)

else:
    scaler = StandardScaler()
    X_scaled = scaler.fit_transform(X)
    model = SVR(kernel='rbf')
    model.fit(X_scaled, y)
    y_pred = model.predict(X_scaled)
    user_input_scaled = scaler.transform([[age, height, weight]])
    predicted_bmi = model.predict(user_input_scaled)[0]
    accuracy = r2_score(y, y_pred)

# 📈 Results Display
st.subheader("Results")
st.write(f"📏 **Actual BMI (based on formula):** {bmi:.2f}")
st.write(f"🤖 **Predicted BMI using {model_choice}:** {predicted_bmi:.2f}")
st.write(f"🎯 **Model Accuracy (R² Score):** {accuracy*100:.2f}%")

if bmi < 18.5:
    category = "Underweight"
    color = "lightblue"
elif 18.5 <= bmi < 24.9:
    category = "Normal weight"
    color = "lightgreen"
elif 25 <= bmi < 29.9:
    category = "Overweight"
    color = "orange"
else:
    category = "Obese"
    color = "red"

st.markdown(f"<h4 style='color:{color};'>🏷️ Category: {category}</h4>", unsafe_allow_html=True)

# 🪄 Footer Note
st.info("Switch between Random Forest and SVM to compare accuracy and predictions. This app is for
educational purposes only.")
```


BMI prediction app
BMI Prediction
bmi
bmi
BMI Prediction App
localhost:8501
Deploy
BMI Prediction App
Predict your Body Mass Index (BMI) using Random Forest or SVM, and view model accuracy.
Enter Your Details:
Age (in years)
20
Height (in cm)
160
Weight (in kg)
60
Select Machine Learning Model
Random Forest Regressor
Results
Actual BMI (based on formula): 23.44
Predicted BMI using Random Forest Regressor: 23.43
30°C
Partly cloudy
Search
ENG
IN
06:21 PM
29-10-2025


BMI prediction app
BMI Prediction
bmi
bmi
BMI Prediction App
localhost:8501
160
Deploy
Weight (in kg)
60
Select Machine Learning Model
Random Forest Regressor
Results
Actual BMI (based on formula): 23.44
Predicted BMI using Random Forest Regressor: 23.43
Model Accuracy (R² Score): 99.77%
Category: Normal weight
Switch between Random Forest and SVM to compare accuracy and predictions. This app is for educational purposes only.
30°C
Partly cloudy
Search
ENG
IN
06:21 PM
29-10-2025

RESULT:

The Linear Regression model was successfully trained and used to predict stock prices. The predicted values closely followed the actual prices, showing that the model effectively captured the overall trend of the stock data.