Code:

```
import pandas as pd
import pymongo
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
csv_file = "C:/Users/HP/Downloads/stock_data_200 (2).csv"
data = pd.read_csv(csv_file)
print("CSV Data Loaded Successfully!")
print(data.head())
plt.figure()
plt.plot(data["Close"])
plt.title("Stock Closing Price Trend")
plt.xlabel("Days")
plt.ylabel("Closing Price")
plt.show()

plt.figure()
plt.bar(range(len(data)), data["Volume"])
plt.title("Stock Volume Traded")
plt.xlabel("Days")
plt.ylabel("Volume")
plt.show()
plt.figure()
corr = data[["Open", "High", "Low", "Close", "Volume"]].corr()
sns.heatmap(corr, annot=True)
plt.title("Feature Correlation Heatmap")
plt.show()
client = pymongo.MongoClient("mongodb://localhost:27017/")
db = client["stockDB"]
```

```
collection = db["stockPrices"]
collection.delete_many({})
records = data.to_dict(orient="records")
collection.insert_many(records)
print("Data inserted into MongoDB successfully!")
fetched_data = pd.DataFrame(list(collection.find()))
print("\nData fetched from MongoDB:")
print(fetched_data.head())
X = fetched_data[["Open", "High", "Low", "Close", "Volume"]]
y = fetched_data["Trend"]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
print("\nModel Training Completed!")
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("\nModel Accuracy:", accuracy)
print("\nConfusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))
plt.figure()
importances = model.feature_importances_
features = X.columns
plt.bar(features, importances)
plt.title("Feature Importance in Random Forest")
plt.xlabel("Features")
plt.ylabel("Importance")
plt.show()
example = pd.DataFrame({
    "Open": [150],
    "High": [152],
    "Low": [149],
    "Close": [151],
    "Volume": [1200000]
})
example_pred = model.predict(example)
print("\nPredicted Trend for Example Input:", example_pred[0])
print("\nProject Execution Completed Successfully!")
```