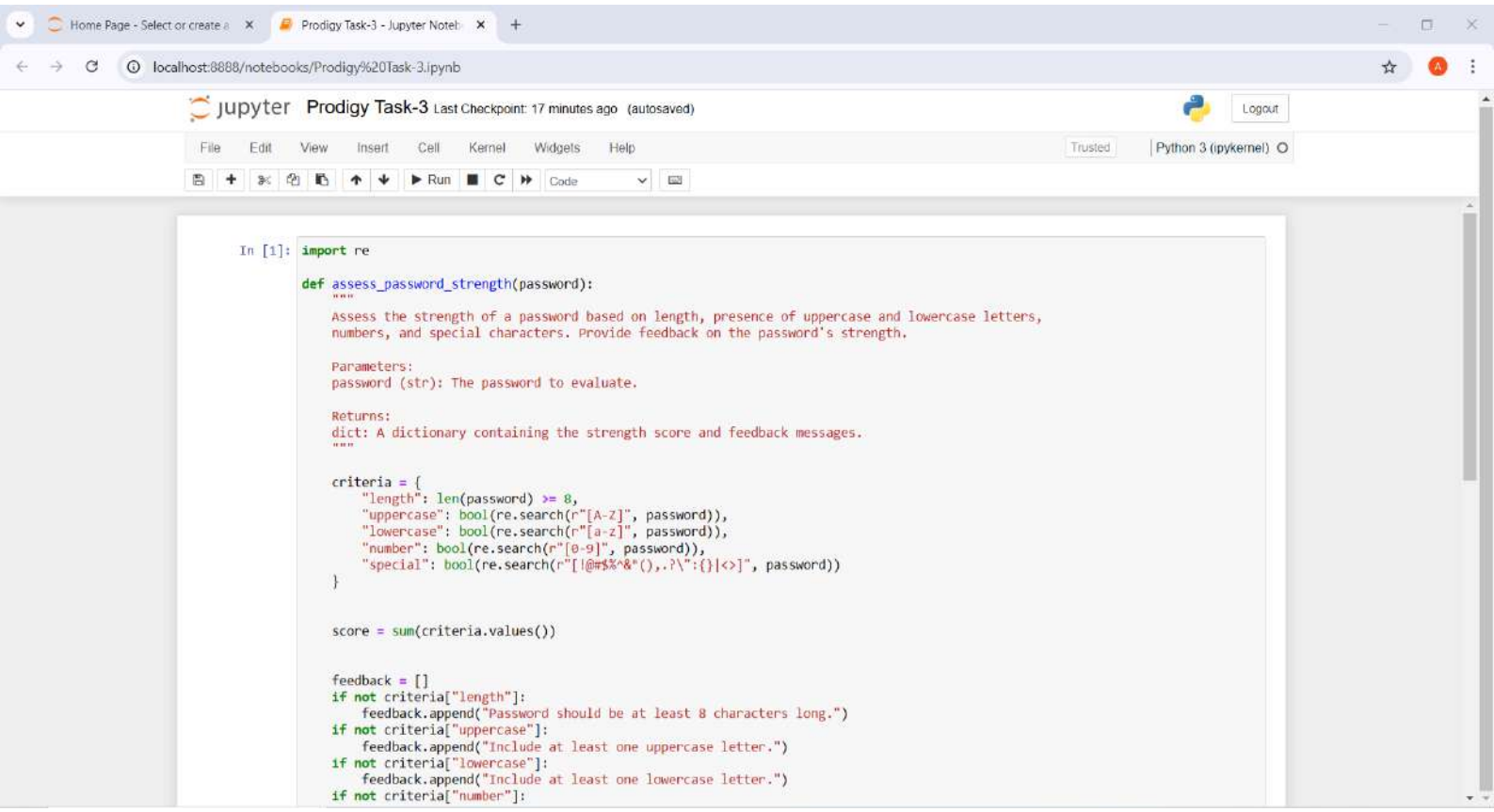
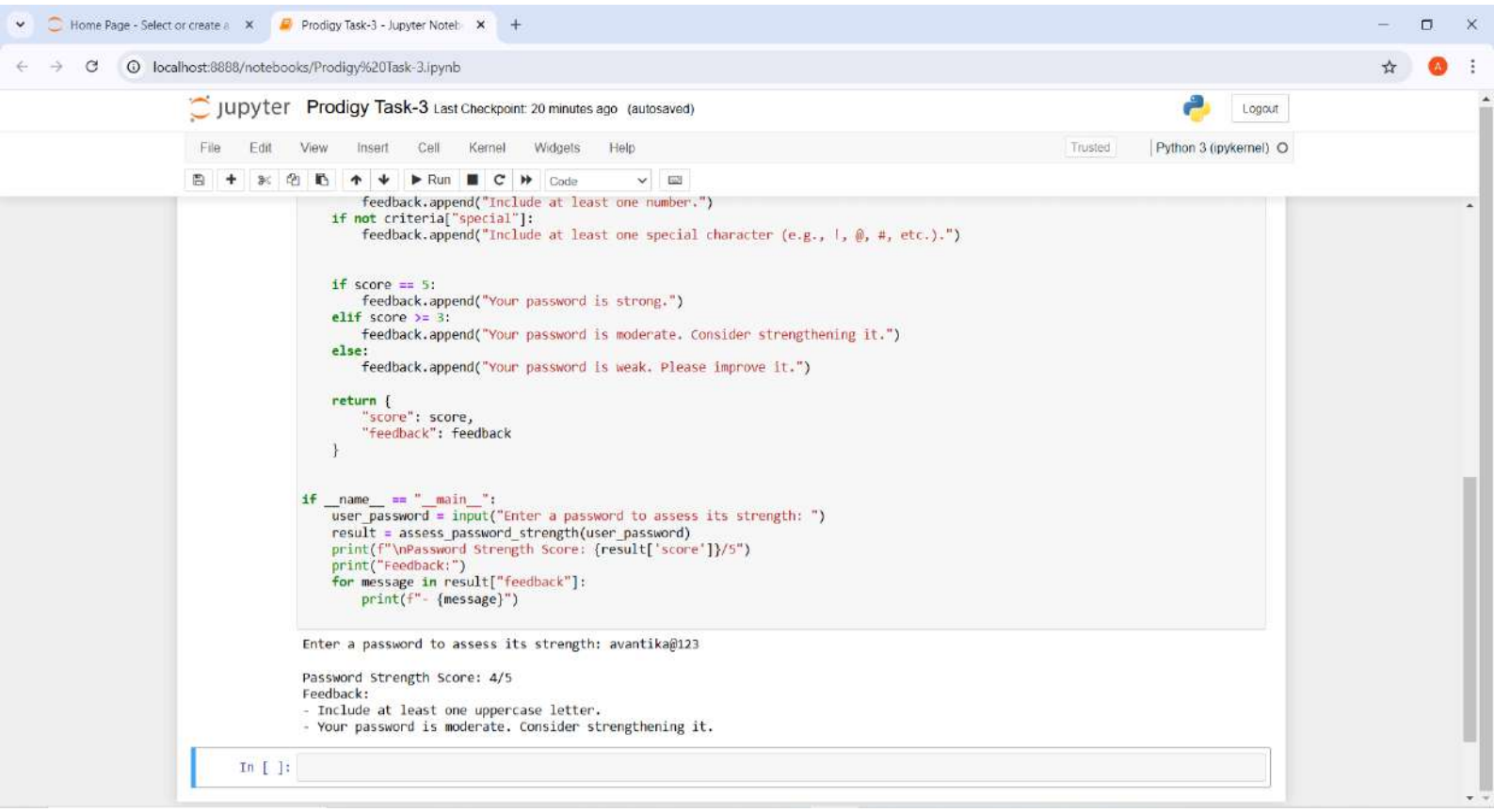

Home Page - Select or create a
Prodigy Task-3 - Jupyter Noteb
localhost:8888/notebooks/Prodigy%20Task-3.ipynb
jupyter Prodigy Task-3 Last Checkpoint: 17 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
In [1]:
import re
def assess_password_strength(password):
"""
Assess the strength of a password based on length, presence of uppercase and lowercase letters,
numbers, and special characters. Provide feedback on the password's strength.
Parameters:
password (str): The password to evaluate.
Returns:
dict: A dictionary containing the strength score and feedback messages.
"""
criteria = {
"length": len(password) >= 8,
"uppercase": bool(re.search(r"[A-Z]", password)),
"lowercase": bool(re.search(r"[a-z]", password)),
"number": bool(re.search(r"[0-9]", password)),
"special": bool(re.search(r"[!@#$%^&*(),.?\":{}|<>]", password))
}
score = sum(criteria.values())
feedback = []
if not criteria["length"]:
feedback.append("Password should be at least 8 characters long.")
if not criteria["uppercase"]:
feedback.append("Include at least one uppercase letter.")
if not criteria["lowercase"]:
feedback.append("Include at least one lowercase letter.")
if not criteria["number"]:


Home Page - Select or create a
Prodigy Task-3 - Jupyter Noteb
localhost:8888/notebooks/Prodigy%20Task-3.ipynb
jupyter Prodigy Task-3 Last Checkpoint: 20 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
feedback.append("Include at least one number.")
if not criteria["special"]:
feedback.append("Include at least one special character (e.g., !, @, #, etc.).")
if score == 5:
feedback.append("Your password is strong.")
elif score >= 3:
feedback.append("Your password is moderate. Consider strengthening it.")
else:
feedback.append("Your password is weak. Please improve it.")
return {
"score": score,
"feedback": feedback
}
if __name__ == "__main__":
user_password = input("Enter a password to assess its strength: ")
result = assess_password_strength(user_password)
print(f"\nPassword Strength Score: {result['score']}/5")
print("Feedback:")
for message in result["feedback"]:
print(f"- {message}")
Enter a password to assess its strength: avantika@123
Password Strength Score: 4/5
Feedback:
- Include at least one uppercase letter.
- Your password is moderate. Consider strengthening it.
In [ ]: