

```

1  import json
2  from datetime import datetime, timedelta
3
4  class SocialSkillsDevelopment:
5      def __init__(self, name):
6          self.name = name
7          self.skills = {
8              'Active Listening': {'level': 0, 'goals': [], 'progress_log': []},
9              'Empathy': {'level': 0, 'goals': [], 'progress_log': []},
10             'Communication': {'level': 0, 'goals': [], 'progress_log': []},
11             'Emotional Intelligence': {'level': 0, 'goals': [], 'progress_log': []},
12             'Conflict Resolution': {'level': 0, 'goals': [], 'progress_log': []}
13         }
14         self.personal_development_plan = []
15
16     def self_assess(self, skill, level):
17         """
18         Self-assess a specific social skill on a scale of 0-10
19         Args:
20             skill (str): The social skill to assess
21             level (int): Skill level from 0-10
22         """
23         if skill in self.skills and 0 ≤ level ≤ 10:
24             previous_level = self.skills[skill]['level']
25             self.skills[skill]['level'] = level
26
27             # Log the assessment
28             assessment_entry = {
29                 'date': datetime.now().strftime("%Y-%m-%d"),
30                 'previous_level': previous_level,
31                 'new_level': level
32             }
33             self.skills[skill]['progress_log'].append(assessment_entry)
34             print(f"{skill} skill level updated to {level}")
35         else:
36             print("Invalid skill or level. Please choose a valid skill and level
37 between 0-10.")
38
39     def set_skill_goal(self, skill, goal_description, target_level, target_date):
40         """
41         Set a specific goal for a social skill
42         Args:
43             skill (str): The social skill to set a goal for
44             goal_description (str): Detailed description of the goal
45             target_level (int): Target skill level
46             target_date (str): Target date to achieve the goal (YYYY-MM-DD)
47         """
48         if skill in self.skills and 0 ≤ target_level ≤ 10:
49             goal = {
50                 'description': goal_description,

```

```

50         'target_level': target_level,
51         'target_date': target_date,
52         'status': 'In Progress',
53         'created_date': datetime.now().strftime("%Y-%m-%d")
54     }
55     self.skills[skill]['goals'].append(goal)
56     print(f"Goal set for {skill}: {goal_description}")
57 else:
58     print("Invalid skill or target level.")
59
60 def track_goal_progress(self, skill, goal_index):
61     """
62     Track progress on a specific goal
63     Args:
64         skill (str): The social skill
65         goal_index (int): Index of the goal in the goals list
66     """
67     if skill in self.skills and 0 ≤ goal_index < len(self.skills[skill]
['goals']):
68         goal = self.skills[skill]['goals'][goal_index]
69         current_level = self.skills[skill]['level']
70
71         if current_level ≥ goal['target_level']:
72             goal['status'] = 'Completed'
73             print(f"Congratulations! Goal completed: {goal['description']}")
74         else:
75             days_remaining = (datetime.strptime(goal['target_date'], "%Y-%m-%d")
- datetime.now()).days
76             progress_percentage = (current_level / goal['target_level']) * 100
77
78             print(f"Goal Progress for {skill}:")
79             print(f"Description: {goal['description']}")
80             print(f"Current Level: {current_level}")
81             print(f"Target Level: {goal['target_level']}")
82             print(f"Progress: {progress_percentage:.2f}%")
83             print(f"Days Remaining: {days_remaining}")
84
85 def generate_development_report(self):
86     """
87     Generate a comprehensive personal development report
88     """
89     print(f"\n--- Social Skills Development Report for {self.name} ---")
90     for skill, data in self.skills.items():
91         print(f"\n{skill}:")
92         print(f"Current Level: {data['level']}/10")
93
94         if data['goals']:
95             print("Active Goals:")
96             for idx, goal in enumerate(data['goals']):
97                 print(f" {idx + 1}. {goal['description']} (Status:
{goal['status']}")

```

```

98
99         if data['progress_log']:
100             print("Recent Progress:")
101             for log in data['progress_log'][-3:]: # Show last 3 progress
entries
102                 print(f" {log['date']}: Level changed from
{log['previous_level']} to {log['new_level']}")
103
104 def main():
105     # Example usage
106     user = SocialSkillsDevelopment("Alex")
107
108     # Initial self-assessment
109     user.self_assess("Communication", 4)
110
111     # Set goals
112     user.set_skill_goal(
113         "Communication",
114         "Improve public speaking confidence",
115         8,
116         (datetime.now() + timedelta(days=180)).strftime("%Y-%m-%d")
117     )
118
119     # Simulate progress
120     user.self_assess("Communication", 6)
121     user.track_goal_progress("Communication", 0)
122
123     # Generate report
124     user.generate_development_report()
125
126 if __name__ == "__main__":
127     main()

```