```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

# Sample data for training a placeholder model
data = {
    'age': [15, 18, 20, 22, 25, 27, 30, 32, 35, 40],
    'height': [150, 160, 165, 170, 175, 180, 182, 185, 170, 168],
    'weight': [50, 60, 65, 70, 75, 80, 82, 85, 90, 95],
    'skill_level': [0, 1, 1, 2, 2, 0, 1, 2, 0, 1]  # 0: Beginner, 1: Intermediate, 2: Expe
}

# Convert data to DataFrame
df = pd.DataFrame(data)
X = df[['age', 'height', 'weight']]
y = df['skill_level']

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train the Random Forest model
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)

# Define recommendations based on skill level
recommendations = {
    "Beginner": {
        "Shots to Learn": [
            "Front Foot Defense: Basic forward defensive shot to block the ball safely.",
            "Back Foot Defense: Defensive shot on the back foot for short-pitched deliveri
            "Straight Drive: Controlled shot along the ground.",
            "Pull Shot: For balls on a shorter length, teaching wrist and shoulder positio
        ],
        "Guidance": [
            "Focus on stance and grip basics, ensuring a solid foundation.",
            "Introduce drills to improve hand-eye coordination.",
            "Practice in nets with slow bowling to build confidence."
        ]
    },
    "Intermediate": {
        "Shots to Learn": [
            "Cover Drive: Emphasize body alignment, high elbow, and wrist positioning.",
            "Square Cut: Ideal for shorter deliveries outside off-stump.",
            "On Drive: Requires precise footwork and timing.",
            "Hook Shot: For bouncers; focus on head positioning and follow-through."
        ],
        "Guidance": [
            "Focus on dynamic footwork drills to enhance adaptability.",
            "Practice with medium-speed bowling, incorporating more diverse bowling angles
            "Emphasize shot selection based on ball placement."
        ]
    },
    },
    "Expert": {
        "Shots to Learn": [
            "Late Cut: Delicate shot requiring control and precise timing.",
            "Scoop or Ramp Shot: Used against pace bowlers, especially in T20s.",
            "Reverse Sweep: For spinners, requires excellent timing and balance.",
            "Upper Cut: For short balls close to the body."
        ],
        "Guidance": [
            "Practice adaptability to different pitch types and pace variations.",
            "Train with varying ball speeds and bowling styles.",
            "Focus on situational practice to improve shot selection."
        ]
    }
}

def get_user_input():
    # Prompt the user for personal details
    age = int(input("Enter your age: "))
    height = int(input("Enter your height (in cm): "))
    weight = int(input("Enter your weight (in kg): "))

    # Prompt for cricket skill level
    print("\nSelect your cricket playing ability level:")
    print("1 - Beginner\n2 - Intermediate\n3 - Expert")
    skill_level_choice = int(input("Enter the number corresponding to your skill level: ")

    # Map number to skill level
    skill_levels = {1: "Beginner", 2: "Intermediate", 3: "Expert"}
    skill_level = skill_levels.get(skill_level_choice, "Beginner")  # Default to Beginner

    return age, height, weight, skill_level

def display_recommendations(skill_level):
    # Display recommendations based on skill level
    rec = recommendations[skill_level]

    print(f"\nSkill Level: {skill_level}\n")
    print("Shots to Learn:")
    for shot in rec["Shots to Learn"]:
        print(f"- {shot}")
    print("\nGuidance:")
    for guide in rec["Guidance"]:
        print(f"- {guide}")

# Run the program
age, height, weight, skill_level = get_user_input()

# Optionally predict skill level (if no direct user input were provided)
# For example, uncomment the following line to use prediction:
# skill_level = model.predict([[age, height, weight]])[0]

display_recommendations(skill_level)
```

```
Enter your age: 21
Enter your height (in cm): 172
Enter your weight (in kg): 62

Select your cricket playing ability level:
1 - Beginner
2 - Intermediate
3 - Expert
Enter the number corresponding to your skill level: 2

Skill Level: Intermediate

Shots to Learn:
- Cover Drive: Emphasize body alignment, high elbow, and wrist positioning.
- Square Cut: Ideal for shorter deliveries outside off-stump.
- On Drive: Requires precise footwork and timing.
- Hook Shot: For bouncers; focus on head positioning and follow-through.

Guidance:
- Focus on dynamic footwork drills to enhance adaptability.
- Practice with medium-speed bowling, incorporating more diverse bowling angles.
- Emphasize shot selection based on ball placement.
```