

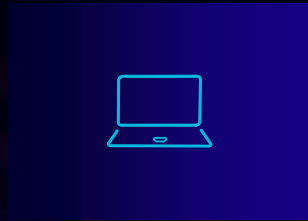


SMART BUSINESS MODEL FOR BADMINTON COURT OPERATIONS
FROM SETUP TO MARKETING

INTRODUCTION TO PYTHON PROGRAMMING SS 2025

PROFESSOR: DR. BABY

Bharath Yadav Jayaram (400885384)
Pavan Chandrashekar Shetty (400887576)
Madhan Gowda Mahadevan (400882767)
Gagan Gowda (400886935)
Skanda Gowda (400956772)



Focus Areas:

- Setup cost planning
- Daily operational expenses
- Booking & availability system
- Financial metrics and growth
- Discounts & marketing features

STUDENT 1 SETUP AND COST PLANNING



```
def calculate_setup_cost():
    print(" Badminton Court Setup Cost Estimator\n")

    # Ask user for number of courts
    try:
        number_of_courts = int(input("Enter the number of badminton courts to set up: "))
        if number_of_courts < 1:
            raise ValueError("Number of courts must be at least 1.")
    except ValueError as e:
        print(f" Invalid input: {e}")
        return

    # Per court material and facility costs
    flooring = 3600          # Synthetic flooring (300 m²)
    poles = 2 * 120          # 2 poles per court
    net = 80                 # 1 net per court
    lighting = 4 * 260       # 4 lights per court
    markings = 180           # Court markings
    labour = 1499            # Labour charges
    chairs = 2 * 45          # 2 chairs per court
    fan = 40                 # 1 fan per court
    reception = 76          # 1 reception desk per court

    # Total per court
    per_court_total = flooring + poles + net + lighting + markings + labour + chairs + fan + reception
    overall_total = per_court_total * number_of_courts

    # Print breakdown
    print("\n Cost Breakdown Per Court:")
    print(f" Flooring: €{flooring}")
    print(f" Poles (2): €{poles}")
    print(f" Net (1): €{net}")
    print(f" Lighting (4): €{lighting}")
    print(f" Court Markings: €{markings}")
    print(f" Labour: €{labour}")
    print(f" Chairs (2): €{chairs}")
    print(f" Fan (1): €{fan}")
    print(f" Reception Desk: €{reception}")
    print(f" Total per court: €{per_court_total}")

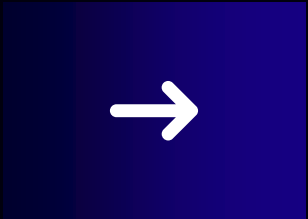
    print("\n Total Setup Cost for All Courts:")
    print(f" Courts: {number_of_courts}")
    print(f" Grand Total: €{overall_total}")

    # Run the function
    calculate_setup_cost()
```

Badminton COURT SETUP COST CALCULATOR CODE STEP BY STEP TO UNDERSTAND WHAT EACH PART DOES AND HOW THE TOTAL COST IS CALCULATED FOR N COURTS.

1. Defines a function called `calculate_setup_cost` that will contain all our logic for asking the user and calculating the total cost.
2. A friendly message for the user, telling them what this program does.
3. Asking the user how many courts they want to set up.
4. Ensures the input is an integer and at least 1.
5. If not, it catches the error and prints a friendly message
6. These are the **fixed material/facility costs per court**.
7. `per_court_total`: sums all individual item costs for **one court**.
8. `overall_total`: multiplies that by the number of courts (n) to get **total cost**.
9. prints each cost item separately, helping the user understand how the total was built.
10. Shows how many courts were calculated and what the **final total setup cost** is.
11. Calls the function so the program starts running.
12. Get a full estimate for setting up **1 or more courts**
13. See cost breakdown per item
14. Ensure clean user input (only allows numbers ≥ 1)





INPUT

Cost Breakdown Per Court:

- Flooring: €3600
- Poles (2): €240
- Net (1): €80
- Lighting (4): €1040
- Court Markings: €180
- Labor: €1499
- Chairs (2): €90
- Fan (1): €40
- Reception Desk: €76

OUTPUT

Badminton Court Setup Cost Estimator
Enter the number of badminton courts to set up: 7
Cost Breakdown Per Court: Flooring: €3600
Poles (2): €240
Net (1): €80
Lighting (4): €1040
Court Markings: €180
Labour: €1499
Chairs (2): €90
Fan (1): €40
Reception Desk: €76
Total per court: €6845
Total Setup Cost for All Courts: Courts: 7
Grand Total: €47915

```
➡ Badminton Court Setup Cost Estimator

Enter the number of badminton courts to set up: 7

Cost Breakdown Per Court:
Flooring: €3600
Poles (2): €240
Net (1): €80
Lighting (4): €1040
Court Markings: €180
Labour: €1499
Chairs (2): €90
Fan (1): €40
Reception Desk: €76
Total per court: €6845

Total Setup Cost for All Courts:
Courts: 7
Grand Total: €47915
```



STUDENT 2: FACILITY MANAGEMENT & OPERATIONS

```
def calculate_monthly_costs():
    print(" Welcome to the Badminton Court Monthly Cost Calculator\n")

    try:
        # Input: number of courts
        num_courts = int(input("Enter the number of badminton courts: "))
        if num_courts < 1:
            raise ValueError("Number of courts must be at least 1.")

        # Input: staff counts
        receptionists = int(input("Enter the number of Receptionists: "))
        trainees = int(input("Enter the number of Trainees: "))
        cleaners = int(input("Enter the number of Cleaners: "))

        if any(n < 0 for n in [receptionists, trainees, cleaners]):
            raise ValueError("Staff counts cannot be negative.")

    except ValueError as e:
        print(f" Invalid input: {e}")
        return

    # Salaries per person
    salary_per_receptionist = 1600
    salary_per_trainee = 2250
    salary_per_cleaner = 600

    # Total staff cost
    total_receptionist_cost = receptionists * salary_per_receptionist
```

```
total_trainee_cost = trainees * salary_per_trainee
total_cleaner_cost = cleaners * salary_per_cleaner
total_staff_cost = total_receptionist_cost + total_trainee_cost + total_cleaner_cost
```

```
# Fixed monthly costs per court
```

```
maintenance_per_court = 240
```

```
energy_per_court = 799
```

```
# Scaled by number of courts
```

```
total_maintenance_cost = maintenance_per_court * num_courts
```

```
total_energy_cost = energy_per_court * num_courts
```

```
# Total monthly operational cost
```

```
total_monthly_cost = total_staff_cost + total_maintenance_cost + total_energy_cost
```

```
total_staff = receptionists + trainees + cleaners
```

```
# Output
```

```
print("\n Monthly Operational Cost Report")
```

```
print(f" Number of courts: {num_courts}")
```

```
print(f" Total staff members: {total_staff}")
```

```
print(" Staff Breakdown:")
```

```
print(f" - Receptionists: {receptionists} staff, €{total_receptionist_cost}")
```

```
print(f" - Trainees: {trainees} staff, €{total_trainee_cost}")
```

```
print(f" - Cleaners: {cleaners} staff, €{total_cleaner_cost}")
```

```
print(f"\n Staff cost total: €{total_staff_cost}")
```

```
print(f" Maintenance cost (€{maintenance_per_court} x {num_courts}): €{total_maintenance_cost}")
```

```
print(f" Energy cost (€{energy_per_court} x {num_courts}): €{total_energy_cost}")
```

```
print(f"\n Total Monthly Operational Cost: €{total_monthly_cost}")
```

```
# Run the function
```

```
calculate_monthly_costs()
```


STEP BY STEP TO UNDERSTAND WHAT EACH PART DOES AND HOW THE TOTAL MONTHLY OPERATIONAL COST.

1. Asks the user to input:
Number of courts
Number of each staff type (Receptionists, Trainees, Cleaners)
Sets monthly **salary per person** for each staff role.
2. Calculates total monthly salary paid for each staff type.
3. Adds them to get the **total staff salary**
4. Sets fixed **maintenance and energy costs** per court.
5. Multiplies by number of courts to get total cost.
6. Adds **staff cost + maintenance + energy** to get total monthly expenses
7. Displays a clear report:
8. Number of courts
9. Staff count and individual costs
10. Maintenance and energy costs
11. **Total monthly cost**



INPUT

Enter the number of badminton courts:

Enter the number of Receptionists:

Enter the number of Trainees:

Enter the number of Cleaners:

Staff Salaries (per month per person)

Receptionist: €1600

Trainee: €2250

Cleaner: €600

Fixed Costs (per court, per month)

Maintenance: €240

Energy (lights, fans, water): €799

OUTPUT

Welcome to the Badminton Court Monthly Cost Calculator

Enter the number of badminton courts: 10

Enter the number of Receptionists: 20

Enter the number of Trainees: 10

Enter the number of Cleaners: 20 Monthly Operational Cost Report

Number of courts: 10

Total staff members: 50

Staff Breakdown: -

- Receptionists: 20 staff, €32000

- Trainees: 10 staff, €22500

- Cleaners: 20 staff, €12000

- Staff cost total: €66500

- Maintenance cost (€240 x 10): €2400

- Energy cost (€799 x 10): €7990

- Total Monthly Operational Cost: €76890





OUTPUT



Welcome to the Badminton Court Monthly Cost Calculator

Enter the number of badminton courts: 10

Enter the number of Receptionists: 20

Enter the number of Trainees: 10

Enter the number of Cleaners: 20

Monthly Operational Cost Report

Number of courts: 10

Total staff members: 50

Staff Breakdown:

- Receptionists: 20 staff, €32000
- Trainees: 10 staff, €22500
- Cleaners: 20 staff, €12000

Staff cost total: €66500

Maintenance cost (€240 x 10): €2400

Energy cost (€799 x 10): €7990

Total Monthly Operational Cost: €76890

STUDENT 3: BOOKING AND AVAILABILITY

```
✓ 21s ▶ def display_schedule():
    print("\n Court Availability:")
    print("Open daily from 10:00 AM to 11:00 PM (13 hours per day)")

def get_hourly_rate(day_type):
    day_type = day_type.lower()
    if day_type in ["saturday", "sunday"]:
        return 7.0 # Weekend rate
    else:
        return 4.5 # Weekday rate

def calculate_booking_cost(day_type, hours_booked):
    rate = get_hourly_rate(day_type)
    return rate * hours_booked

def value_added_services(coaching_hours, rackets_rented, memberships):
    coaching_rate = 5.0 # Per hour per person
    racket_rate = 1.0 # Per racket
    membership_rate = 79.0 # Per month
    total_coaching = coaching_hours * coaching_rate
    total_rackets = rackets_rented * racket_rate
    total_membership = memberships * membership_rate
    total_service_cost = total_coaching + total_rackets + total_membership
    return total_service_cost
```

```
✓ 21s ▶ # --- Program starts here ---
display_schedule()

# Input booking details
day_type = input("\nEnter the day (e.g., Monday, Saturday): ")
hours_booked = int(input("Enter number of hours to book the court (max 13): "))

# Validate hour range
if hours_booked < 1 or hours_booked > 13:
    print("Invalid number of hours. Please enter a value between 1 and 13.")
    exit()

# Calculate court rental cost
booking_cost = calculate_booking_cost(day_type, hours_booked)
print(f"\n Court rental cost on {day_type.capitalize()}: €{booking_cost:.2f}")

# Input value-added services
print("\n Value Added Services")
coaching_hours = int(input("Enter coaching hours (per person): "))
rackets_rented = int(input("Enter number of rackets to rent: "))
memberships = int(input("Enter number of memberships to purchase: "))

# Calculate service cost
services_cost = value_added_services(coaching_hours, rackets_rented, memberships)

# Total cost summary
total = booking_cost + services_cost
print("\n Cost Summary")
print(f"Court rental: €{booking_cost:.2f}")
print(f"Value added services: €{services_cost:.2f}")
print(f"Total cost for booking + services: €{total:.2f}")
```



STEP BY STEP TO UNDERSTAND WHAT EACH PART DOES AND HOW THE TOTAL COST OF ALL SELECTED SERVICES

What This Program Does:

Allows users to:

Book a badminton court

Add extra services like coaching, racket rental, and memberships

View cost summary based on day and services

Prints opening hours: 10:00 AM to 11:00 PM (13 hours)

1. Weekday rate: €4.50/hour
2. Weekend rate: €7.00/hour
3. Multiplies the hourly rate by number of hours booked
4. Coaching: €5.00/hour/person
5. Racket Rental: €1.00 each
6. Membership: €79.00/month
7. Calculates total cost of all selected services
8. Asks user for booking day and hours

Displays:

Court rental cost

Extra services cost

Total combined cost



INPUT

Enter the day (e.g., Monday, Saturday):

Enter number of hours to book the court (max 13):

Enter coaching hours (per person):

Enter number of rackets to rent:

Enter number of memberships to purchase:

Weekday rate: €4.50/hour

Weekend rate: €7.00/hour

Coaching: €5.00/hour/person

Racket Rental: €1.00 each

Membership: €79.00/month

Calculates total cost of all selected services

OUTPUT

Court Availability: Open daily from 10:00 AM to 11:00 PM (13 hours per day)

Enter the day (e.g., Monday, Saturday): M

Enter number of hours to book the court (max 13): 10

Court rental cost on M: €45.00

Value Added Services Enter coaching hours (per person): 0

Enter number of rackets to rent: 16

Enter number of memberships to purchase: 0

Cost Summary

Court rental: €45.00

Value added services: €16.00

Total cost for booking + services: €61.00





OUTPUT



```
Court Availability:  
Open daily from 10:00 AM to 11:00 PM (13 hours per day)  
  
Enter the day (e.g., Monday, Saturday): M  
Enter number of hours to book the court (max 13): 10  
  
Court rental cost on M: €45.00  
  
Value Added Services  
Enter coaching hours (per person): 0  
Enter number of rackets to rent: 16  
Enter number of memberships to purchase: 0  
  
Cost Summary  
Court rental: €45.00  
Value added services: €16.00  
Total cost for booking + services: €61.00
```



STUDENT 4: FINANCE AND GROWTH STRATEGY

```
import matplotlib.pyplot as plt

def calculate_business_metrics(monthly_revenue, setup_cost, monthly_operational_cost):
    # 1. Calculate monthly profit
    monthly_profit = monthly_revenue - monthly_operational_cost
    print(f"Monthly Revenue: €{monthly_revenue}")
    print(f"Monthly Operational Cost: €{monthly_operational_cost}")
    print(f"Monthly Profit: €{monthly_profit}")

    # 2. Compare setup and operational cost
    print(f"\nInitial Setup Cost: €{setup_cost}")
    if monthly_profit <= 0:
        print("Warning: Business is not profitable. Break-even is not achievable at current rates.")
        return

    # 3. Calculate break-even point
    break_even_months = setup_cost / monthly_profit
    print(f"\nBreak-even Point: {break_even_months:.2f} months")
    print(f"The initial investment of €{setup_cost} will be recovered in approximately {round(break_even_months)} months.")

    # 4. Plotting
    months = list(range(1, int(round(break_even_months)) + 6)) # Extra 5 months for visualization
    cumulative_profit = [monthly_profit * month for month in months]
```




CODE

```
plt.figure(figsize=(10, 6))
plt.plot(months, cumulative_profit, label='Cumulative Profit (€)', marker='o')
plt.axhline(y=setup_cost, color='r', linestyle='--', label='Setup Cost (€)')
plt.axvline(x=break_even_months, color='g', linestyle='--', label=f'Break-even Point ({break_even_months:.1f} months)')
plt.title('Break-even Analysis and Investment Return')
plt.xlabel('Months')
plt.ylabel('Euros (€)')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()

# --- Input Values ---
monthly_revenue = 14500      # euros
setup_cost = 27380          # euros
monthly_operational_cost = 10539 # euros
```

• It tells how long (in months) until the business covers its initial setup cost.



Badminton **STEP BY STEP TO UNDERSTAND WHAT EACH PART DOES**

This Python program calculates:

1. **Monthly profit**
2. **Break-even point** (how many months it takes to recover the initial investment)
3. **Returns over time** : It also **visualizes** the cumulative profit vs the setup cost to understand profitability

A function that takes 3 inputs:

- **monthly_revenue**: Total income from bookings/services per month.
- **setup_cost**: One-time investment to build/setup the courts.
- **monthly_operational_cost**: Cost to run the facility every month (staff, electricity, etc.).
- **Simple subtraction**: what's left after paying monthly expenses.

Shows whether the business is earning more than it's spending.

If the profit is zero or negative, you can't recover your setup cost. Program exits early.

- **Formula**: Break-even time = Setup Cost / Monthly Profit
- It tells how long (in months) until the business covers its initial setup cost.
- Shows estimated time to reach break-even (rounded and actual decimal).
- Generates a list of months for the **X-axis**.
- Calculates cumulative profit each month for plotting.
- Adds labels, title, legend, and shows the plot.

INPUT

monthly_revenue = 14500
setup_cost = 27380
monthly_operational_cost = 10539

OUTPUT

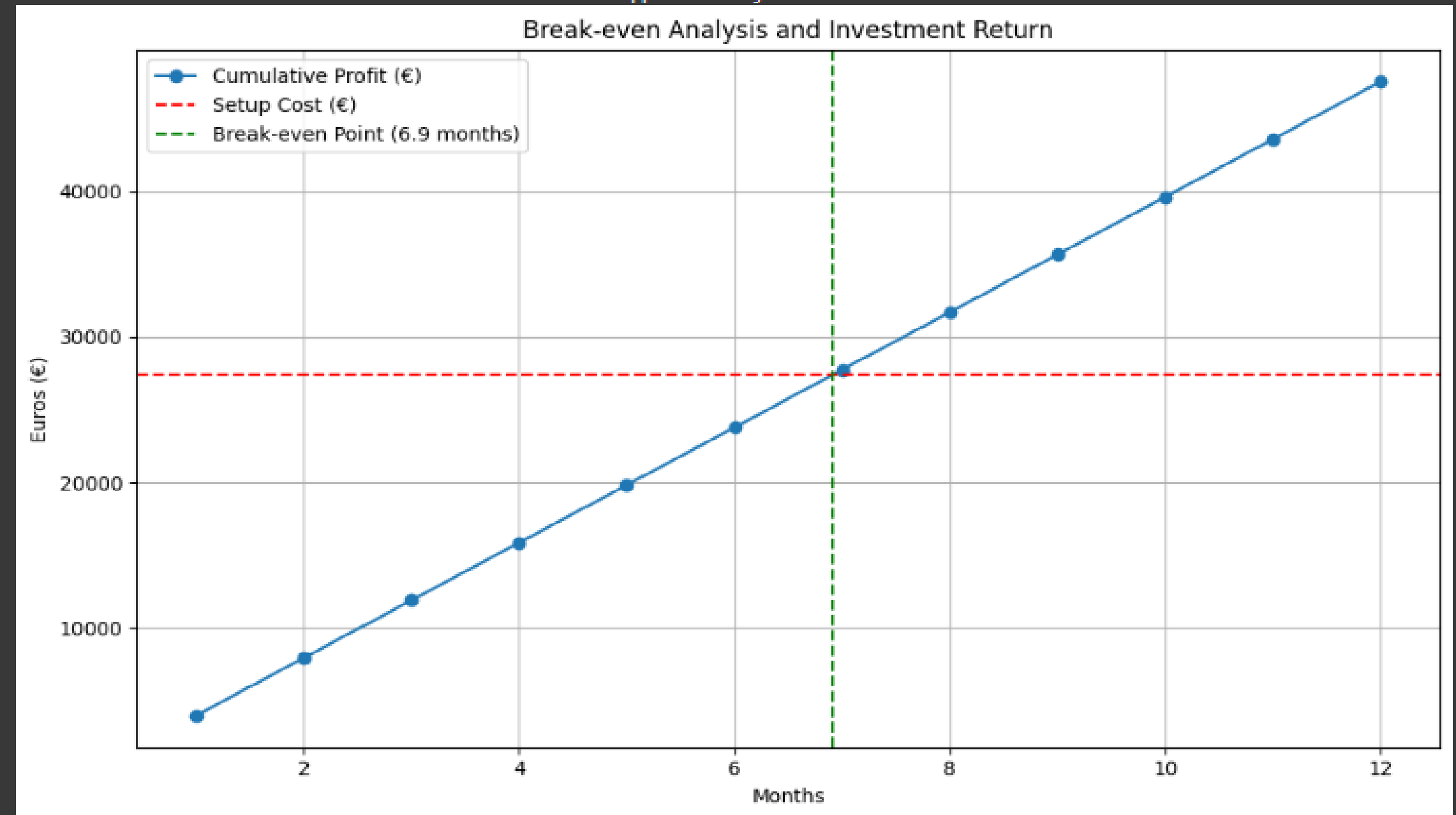
Monthly Revenue: €14500
Monthly Operational Cost: €10539
Monthly Profit: €3961
Initial Setup Cost: €27380
Break-even Point: 6.91 months.
The initial investment of €27380 will be recovered in approximately 7 months.

Monthly Revenue: €14500
Monthly Operational Cost: €10539
Monthly Profit: €3961

Initial Setup Cost: €27380

Break-even Point: 6.91 months

The initial investment of €27380 will be recovered in approximately 7 months.



STUDENT 5: DAILY BOOKINGS RATIO AND MARKETING

```
def badminton_booking():
    print("Welcome to Badminton Court Booking System\n")

    # Check if first time
    first_time = input("Is this your first time booking? (yes/no): ").lower()
    if first_time == "yes":
        print(" Congratulations! Your first game is FREE! Total cost: €0.00")
        return

    # Get booking hours
    hours = int(input("How many hours do you want to book? (1 to 13): "))
    time_input = input("What time will your booking start? (e.g., 12PM, 3PM, 6PM or just 12, 3, 6): ").upper()

    # Normalize time input to format like "12PM"
    if time_input.endswith("PM") or time_input.endswith("AM"):
        time = time_input
    else:
        time = time_input + "PM" # Default to PM as courts are open 10AM to 11PM

    # Get student and group booking info
    student = input("Are you a student? (yes/no): ").lower()
    group = input("Is this a group booking (all 4 courts)? (yes/no): ").lower()

    # Check if off-peak hour
    least_occupied_hours = ["12PM", "1PM", "2PM", "3PM", "4PM"]
```





CODE

```
# Determine cost per hour
if time in least_occupied_hours:
    cost_per_hour = 2.5
    print(" Off-peak booking: €2.50 per hour (12PM to 4PM)")
elif group == "yes":
    cost_per_hour = 13.99
    print(" Group booking deal applied: €13.99 per hour for all 4 courts")
elif student == "yes":
    cost_per_hour = 2.5
    print(" Student discount applied: €2.50 per hour")
else:
    cost_per_hour = 4.5
    print(" Standard rate applied: €4.50 per hour")

# Calculate total
total_cost = cost_per_hour * hours

# Print summary
print(f"\n Booking Summary:")
print(f"Start Time: {time}")
print(f"Hours Booked: {hours}")
print(f"Total Cost: €{total_cost:.2f}")

# Run the booking system
badminton_booking()
```

• It tells how long (in months) until the business covers its initial setup cost.



Badminton **STEP BY STEP TO UNDERSTAND WHAT EACH PART DOES**

Purpose: This smart booking system gives custom pricing based on user conditions and helps promote better court usage during less busy hours.

A user-friendly **Badminton Court Booking System** that adjusts pricing based on:

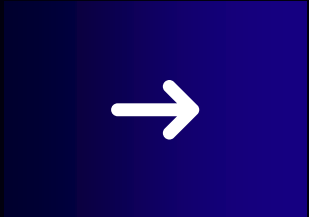
- First-time booking

If yes → first game is **free**; exits the function early.

- Off-peak hours (12PM–4PM)
- Student status
- Group booking (all 4 courts)
- Asks how many hours the user wants to book.
- Asks the start time.
- Makes sure time looks like "12PM", even if user enters just "12".
- Special discounted rate (€2.50/hr) if booked during off-peak hours.
- **Determine Final Hourly Rate:**
- Off-peak → €2.50/hr
- Group booking → €13.99/hr for all 4 courts
- Student → €2.50/hr
- Standard → €4.50/hr

Print Booking Summary:

Shows **start time**, **hours booked**, and **total cost**.



INPUT

Booking Rates:

- First time booking (free game): €0.00 (no charge)
- Off-peak hours (12PM to 4PM): €2.50 per hour
- Student discount: €2.50 per hour
- Group booking (all 4 courts): €13.99 per hour (for all 4 courts)
- Standard rate (all other times): €4.50 per hour

OUTPUT

Output

Welcome to Badminton Court Booking System

Is this your first time booking? (yes/no): no

How many hours do you want to book? (1 to 13): 1

What time will your booking start? (e.g., 12PM, 3PM, 6PM or just 12, 3, 6): 12pm

Are you a student? (yes/no): no

Is this a group booking (all 4 courts)? (yes/no): no

Off-peak booking: €2.50 per hour (12PM to 4PM)

Booking Summary:

Start Time: 12PM Hours

Booked: 1

Total Cost: €2.50



OUTPUT



Welcome to Badminton Court Booking System

Is this your first time booking? (yes/no): NO

How many hours do you want to book? (1 to 13): 1

What time will your booking start? (e.g., 12PM, 3PM, 6PM or just 12, 3, 6): 12PM

Are you a student? (yes/no): NO

Is this a group booking (all 4 courts)? (yes/no): NO

Off-peak booking: €2.50 per hour (12PM to 4PM)

Booking Summary:

Start Time: 12PM

Hours Booked: 1

Total Cost: €2.50



CONCLUSION

- Built as the initial phase of the project
- Can be further developed into a full-featured app
- Scalable for use across India
- Adaptable for expansion to other countries
- Potential to include localized features and multi-language support
- Opportunity to integrate online payments and booking management
- Can enhance user experience with real-time court availability and notifications experience with real-time court availability and notifications



REFERENCES

<https://stackoverflow.com/questions/70088387/python-beginner-help-input-error-handling>

https://www.w3schools.com/python/python_functions.asp

https://www.w3schools.com/python/python_conditions.asp

https://ilias.hs-fresenius.de/ilias.php?baseClass=ilrepositorygui&ref_id=3326828

<https://www.freecodecamp.org/news/python-functions-define-and-call-a-function/>



THANK YOU!

- **Bharath Yadav Jayaram** – jayaram.bharath@stud.hs-fresenius.de
- **Pavan Chandrashekar Shetty** - chandrashekar_shetty.pavan@stud.hs-fresenius.de
- **Madhan Gowda Mahadevan** – mahadevan.madhan@stud.hs-fresenius.de
- **Gagan Gowda** – gowda.gagan@stud.hs-fresenius.de
- **Skanda Gowda** - gowda.skanda@stud.hs-fresenius.de