



Apeejay School
Saket, New Delhi

Apeejay School Saket

Arcade Zone Project Documentation

A Python-based Arcade Management System

Prepared by:

AYUSH RANJAN

AYUSHMAAN GUPTA

Session: 2025-26

Arcade Zone Project Documentation

Index

Section	Page
Introduction	3
Main Application File	4
Functions and Menus	5
Game Logic	6
Database Connection	7
Sample Outputs and Data	8
View All Users (Sample Data)	9
View All Scores (Sample Data)	10
View All Games	11
Player Menu Output	12
Sample Game Outputs	13

Arcade Zone Project Documentation

Introduction

Arcade Zone is a terminal-based arcade management system built in Python. It allows users to register, login, play classic games, and view their scores. Admins can manage users and view all scores. The project demonstrates database integration, menu-driven interfaces, and game logic.

This documentation includes all source code, sample outputs, and data tables for reference.

Arcade Zone Project Documentation

Main Application File (mainframe_1.py)

```
import mysql.connector as m
import os
import sys
import functions as fxn
import db

con = db.con
cur = db.cur

cur.execute("""
    CREATE TABLE IF NOT EXISTS user (
        USER_ID int AUTO_INCREMENT PRIMARY KEY,
        USER_NAME VARCHAR(20) NOT NULL,
        USER_PASSWORD VARCHAR(20) NOT NULL,
        USER_EMAIL VARCHAR(50) NOT NULL,
        USER_MOBILE BIGINT NOT NULL,
        ROLE VARCHAR(40) NOT NULL DEFAULT 'player'
    )
""")
cur.execute("""
    CREATE TABLE IF NOT EXISTS score (
        USER_ID int NOT NULL,
        GAME_ID varchar(20) NOT NULL,
        SCORE_VALUE INT NOT NULL,
        TIME_STAMP TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
        FOREIGN KEY (USER_ID) REFERENCES user(USER_ID)
    )
""")
con.commit()

fxn.welcome_screen()
fxn.mainmenu()

# user admin id = admin
# user admin password = admin@arcade
# user admin email = arcadeworld@arcade.com
```

Arcade Zone Project Documentation

Functions and Menus (functions.py)

[illegible]

Arcade Zone Project Documentation

```
        print("Invalid option, please try again.\n")

def register():
    clear_screen()
    print_banner()
    print("REGISTRATION".center(80))
    print("="*80)
    choice = input("Registration will cause you to share your information with us.\nEnter Y to proceed or N to cancel: ").strip().lower()
    if choice == 'y':
        while True:
            user_name = input("Enter your username (min 3 chars): ").strip()
            if len(user_name) < 3:
                print("Username must be at least 3 characters long.")
                continue
            cur.execute("SELECT * FROM user WHERE USER_NAME = %s", (user_name,))
            if cur.fetchone():
                print("Username already exists. Try another.")
                continue
            break
        while True:
            user_email = input("Enter your email: ").strip()
            if '@' not in user_email or '.' not in user_email:
                print("Invalid email format.")
                continue
            break
        while True:
            mobile = input("Enter your mobile number (10 digits): ").strip()
            if len(mobile) != 10 or not mobile.isdigit():
                print("Mobile number must be 10 digits.")
                continue
            break
        while True:
            password = input("Enter your password: ")
            repassword = input("Re-enter your password: ")
            if password != repassword:
                print("Passwords do not match.")
                continue
            break
        cur.execute(
            "INSERT INTO user (USER_NAME, USER_PASSWORD, USER_EMAIL, USER_MOBILE, ROLE) VALUES (%s, %s, %s, %s, %s)",
            (user_name, password, user_email, mobile, 'player')
        )
        con.commit()
        print("Registration successful!\n")
        input("Press Enter to continue...")
        welcome_screen()
    else:
        print("Registration cancelled.\n")
        input("Press Enter to return to main menu...")
        welcome_screen()
```

Arcade Zone Project Documentation

```
def login():
    clear_screen()
    print_banner()
    print("LOGIN".center(80))
    print("="*80)
    user_name = input("Enter your username: ").strip()
    password = input("Enter your password: ").strip()
    cur.execute("SELECT * FROM user WHERE USER_NAME = %s AND USER_PASSWORD = %s", (user_name,
password))
    user = cur.fetchone()
    if user:
        print(f"\nWelcome {user[1]}!\n")
        if user[5] == 'admin':
            adminmenu()
        elif user[5] == 'player':
            playermenu(user[0])
        else:
            print("Unknown role. Contact admin.")
            mainmenu()
    else:
        print("Invalid username or password. Please try again.")
        input("Press Enter to retry...")
        login()

def adminmenu():
    while True:
        clear_screen()
        print_banner()
        print("ADMIN MENU".center(80))
        print("1. View all users".center(80))
        print("2. View all scores".center(80))
        print("3. View all games".center(80))
        print("4. Suspend a user".center(80))
        print("5. Logout".center(80))
        print("="*80)
        choice = input("Enter your choice (1-5): ").strip()
        if choice == '1':
            clear_screen()
            print_banner()
            cur.execute("SELECT * FROM user")
            users = cur.fetchall()
            print("\nAll Users:")

            print("\n=====")
            print("|| ID || Name || Email || Mobile || Password ||")
            print("|| Role ||")
            print("\n=====")

            for user in users:
```

Arcade Zone Project Documentation

```
        print(f"|| {str(user[0]).ljust(2)} || {user[1].ljust(14)} || {user[3].ljust(26)} ||  
{str(user[4]).ljust(12)} || {user[2].ljust(10)} || {user[5].ljust(8)} ||")  
  
print("┌────────────────────────────────────────────────────────────────────────────────┐  
")  
    input("Press Enter to continue...")  
    elif choice == '2':  
        clear_screen()  
        print_banner()  
        cur.execute("SELECT * FROM score")  
        scores = cur.fetchall()  
        print("\nAll Scores:")  
        if not scores:  
            print("No scores found.")  
        else:  
            print("┌────────────────────────────────────────────────────────────────────────────────┐")  
            print("|| UserID || Game || Score || Time ||")  
            print("└────────────────────────────────────────────────────────────────────────────────┘")  
            for score in scores:  
                print(f"|| {str(score[0]).ljust(6)} || {score[1].ljust(18)} ||  
{str(score[2]).ljust(10)} || {str(score[3]).ljust(22)} ||")  
            print("└────────────────────────────────────────────────────────────────────────────────┘")  
            input("Press Enter to continue...")  
    elif choice == '3':  
        clear_screen()  
        print_banner()  
        print("\nAvailable Games:")  
        print("- Battleship\n- Guess the Number\n- Rock Paper Scissors\n- RPG Battle Arena\n- BlackJack")  
        input("Press Enter to continue...")  
    elif choice == '4':  
        clear_screen()  
        print_banner()  
        user_id = input("Enter the USER_ID of the user to suspend: ").strip()  
        confirm = input(f"Are you sure you want to suspend user {user_id}? (Y/N):  
").strip().lower()  
        if confirm == 'y':  
            if user_id.isdigit():  
                cur.execute("DELETE FROM user WHERE USER_ID = %s", (int(user_id),))  
                cur.execute("DELETE FROM score WHERE USER_ID = %s", (int(user_id),))  
                con.commit()  
                print(f"User with ID {user_id} has been suspended.")  
            else:  
                print("Invalid USER_ID.")  
        else:  
            print("Suspension cancelled.")  
        input("Press Enter to continue...")  
    elif choice == '5':  
        print("Logging out...")  
        welcome_screen()  
        break  
    else:
```


Arcade Zone Project Documentation

```
        print("Invalid choice, please try again.")
        input("Press Enter to continue...")

def playermenu(user_id):
    while True:
        clear_screen()
        print_banner()
        print("PLAYER MENU".center(80))
        print("1. Play Games".center(80))
        print("2. View Scores".center(80))
        print("3. Logout".center(80))
        print("="*80)
        choice = input("Enter your option (1-3): ").strip()
        if choice == '1':
            clear_screen()
            print_banner()
            print("Select a Game".center(80))
            print("1. Battleship".center(80))
            print("2. Guess the Number".center(80))
            print("3. Rock Paper Scissors".center(80))
            print("4. RPG Battle Arena".center(80))
            print("5. BlackJack".center(80))
            print("="*80)
            choice2 = input("Enter your choice (1-5): ").strip()
            if choice2 == '1':
                g.battleship(user_id)
            elif choice2 == '2':
                g.guess_number(user_id)
            elif choice2 == '3':
                g.rock_paper_scissors(user_id)
            elif choice2 == '4':
                g.rpg(user_id)
            elif choice2 == '5':
                g.blackjack(user_id)
            else:
                print("Invalid choice, please try again.")
                input("Press Enter to continue...")
        elif choice == '2':
            clear_screen()
            print_banner()
            cur.execute("SELECT * FROM score WHERE USER_ID = %s", (user_id,))
            scores = cur.fetchall()
            print("\nYour Scores:")
            if not scores:
                print("No scores found.")
            else:
                print("
                |-----|-----|-----|
                | Game       | Score   | Time    |
                |-----|-----|-----|
                ")
                for score in scores:
                    print(f"
                    | {score[1].ljust(18)} | | {str(score[2]).ljust(10)} | |
                    | {str(score[3]).ljust(22)} | |")
```

Arcade Zone Project Documentation

```
        print("┌──────────────────┐┌──────────────────┐┌──────────────────┐")
        input("Press Enter to continue...")
    elif choice == '3':
        print("Logging out...")
        welcome_screen()
        break
    else:
        print("Invalid choice, please try again.")
        input("Press Enter to continue...")
```

Arcade Zone Project Documentation

Game Logic (games.py)

```
import mysql.connector as m
import random
import os
import db

con = db.con
cur = db.cur

def clear_screen():
    os.system("cls" if os.name == "nt" else "clear")

def battleship(user_id):
    clear_screen()
    print("="*60)
    print("BATTLESHIP".center(60))
    print("="*60)
    ship = random.randint(0, 9)
    attempts = 5
    score = 0
    print("Guess the ship's location (0-9)!")

    while attempts > 0:
        try:
            guess = int(input(f"Attempts left {attempts}. Enter your guess: "))
            if guess < 0 or guess > 9:
                print("Please enter a number between 0 and 9.")
                continue
        except ValueError:
            print("Invalid input. Enter a number between 0-9.")
            continue
        if guess == ship:
            print("Hit! You sunk the battleship!")
            score += 5
            break
        else:
            print("Miss!")
            attempts -= 1
    else:
        print("Game over! The ship was at position", ship)
    cur.execute(
        "INSERT INTO score (USER_ID, GAME_ID, SCORE_VALUE) VALUES (%s, %s, %s)",
        (user_id, 'battleship', score)
    )
    con.commit()
    print(f"Score saved! Your score: {score}")
    input("Press Enter to return to menu...")

def guess_number(user_id):
    clear_screen()
```

Arcade Zone Project Documentation

```
print("="*60)
print("GUESS THE NUMBER".center(60))
print("="*60)
number = random.randint(1, 100)
attempts = 7
score = 0
print("Guess the number between 1 and 100!")

while attempts > 0:
    try:
        guess = int(input(f"Attempts left {attempts}. Enter your guess: "))
        if guess < 1 or guess > 100:
            print("Please enter a number between 1 and 100.")
            continue
    except ValueError:
        print("Invalid input. Enter a number between 1-100.")
        continue
    if guess == number:
        print("Congratulations! You guessed the number!")
        score = attempts * 2
        break
    elif guess < number:
        print("Too low!")
    else:
        print("Too high!")
    attempts -= 1
else:
    print(f"Game over! The number was {number}.")
cur.execute(
    "INSERT INTO score (USER_ID, GAME_ID, SCORE_VALUE) VALUES (%s, %s, %s)",
    (user_id, 'guess number', score)
)
con.commit()
print(f"Score saved! Your score: {score}")
input("Press Enter to return to menu...")

def rock_paper_scissors(user_id):
    clear_screen()
    print("="*60)
    print("ROCK PAPER SCISSORS".center(60))
    print("="*60)
    choices = ['rock', 'paper', 'scissors']
    rounds = 5
    score = 0
    for i in range(rounds):
        print(f"Round {i+1} of {rounds}")
        user_choice = input("Choose rock, paper, or scissors: ").strip().lower()
        if user_choice not in choices:
            print("Invalid choice. Try again.")
            continue
        comp_choice = random.choice(choices)
        print(f"Computer chose: {comp_choice}")
```

Arcade Zone Project Documentation

```
        if user_choice == comp_choice:
            print("Draw!")
            score += 1
        elif (user_choice == 'rock' and comp_choice == 'scissors') or \
            (user_choice == 'paper' and comp_choice == 'rock') or \
            (user_choice == 'scissors' and comp_choice == 'paper'):
            print("You win this round!")
            score += 2
        else:
            print("Computer wins this round!")
    print(f"Game Over! Your score: {score}")
    cur.execute(
        "INSERT INTO score (USER_ID, GAME_ID, SCORE_VALUE) VALUES (%s, %s, %s)",
        (user_id, 'rock paper scissors', score)
    )
    con.commit()
    print("Score saved!")
    input("Press Enter to return to menu...")

def rpg(user_id):
    clear_screen()
    print("="*60)
    print("RPG BATTLE ARENA".center(60))
    print("="*60)
    player_hp = 30
    monster_hp = 25
    potions = 3
    monsters = ["Goblin", "Orc", "Troll", "Skeleton", "Zombie", "Vampire", "Werewolf", "Dragon",
                "Slime", "Giant Spider", "Witch", "Demon", "Ghost", "Minotaur", "Hydra", "Golem", "Lich", "Harpy",
                "Basilisk", "Kraken"]
    monsterpickup = random.choice(monsters)
    print(f"A wild {monsterpickup} appears! Prepare for battle!")
    score = 0
    while player_hp > 0 and monster_hp > 0:
        print(f"\nYour HP: {player_hp} | Monster HP: {monster_hp} | Potions: {potions}")
        print("Choose your action:\n1. Attack\n2. Heal\n3. Run")
        choice = input("Enter 1, 2, or 3: ").strip()
        if choice == "1":
            dmg = random.randint(4, 8)
            monster_hp -= dmg
            print(f"You attack the monster for {dmg} damage!")
        elif choice == "2":
            if potions > 0 and player_hp < 70:
                heal = random.randint(6, 12)
                player_hp += heal
                potions -= 1
                print(f"You drink a potion and heal for {heal} HP!")
            elif player_hp > 70:
                print("You are already at full health!")
            else:
                print("You have no potions left!")
        elif choice == "3":
```

Arcade Zone Project Documentation

```
        print("You try to run away...")
        if random.randint(0,5) < 3:
            print("You escaped safely!")
            break
        else:
            print("You failed to escape!")
    else:
        print("Invalid choice. Try again.")
        continue
    if monster_hp > 0:
        monster_dmg = random.randint(3, 7)
        player_hp -= monster_dmg
        print(f"The monster attacks you for {monster_dmg} damage!")
if player_hp <= 0:
    print("\nYou have been defeated by the monster.")
    score -= 5
elif monster_hp <= 0:
    print("\nCongratulations! You defeated the monster!")
    score += 20
elif choice == "3" and player_hp > 0:
    print("\nYou survived by running away!")
cur.execute(
    "INSERT INTO score (USER_ID, GAME_ID, SCORE_VALUE) VALUES (%s, %s, %s)",
    (user_id, 'RPG arena', score)
)
con.commit()
print(f"Score saved! Your score: {score}")
input("Press Enter to return to menu...")

def blackjack(user_id):
    clear_screen()
    print("="*60)
    print("BLACKJACK".center(60))
    print("="*60)
    def deal_card():
        cards = [2, 3, 4, 5, 6, 7, 8, 9, 10, 10, 10, 10, 11]
        return random.choice(cards)
    def calculate_score(hand):
        score = sum(hand)
        aces = hand.count(11)
        while score > 21 and aces:
            score -= 10
            aces -= 1
        return score
    def show_hand(hand, owner="Your"):
        print(f"{owner} cards: {hand} (Total: {calculate_score(hand)})")
    player_hand = [deal_card(), deal_card()]
    dealer_hand = [deal_card(), deal_card()]
    game_over = False
    score = 0
    while not game_over:
        show_hand(player_hand)
```

Arcade Zone Project Documentation

```
print(f"Dealer's first card: {dealer_hand[0]}")
if calculate_score(player_hand) == 21:
    print("Blackjack! You win!")
    score += 20
    game_over = True
    break
elif calculate_score(player_hand) > 21:
    print("You went over 21. You lose!")
    score -= 10
    game_over = True
    break
action = input("Type 'hit' to get another card, or 'stand' to hold: ").lower()
if action == "hit":
    player_hand.append(deal_card())
elif action == "stand":
    break
else:
    print("Invalid input. Please type 'hit' or 'stand'.")
if not game_over:
    while calculate_score(dealer_hand) < 17:
        dealer_hand.append(deal_card())
    show_hand(dealer_hand, "Dealer's")
    player_score = calculate_score(player_hand)
    dealer_score = calculate_score(dealer_hand)
    if dealer_score > 21:
        print("Dealer went over 21. You win!")
        score += 20
    elif dealer_score == player_score:
        print("It's a draw!")
        score += 5
    elif player_score > dealer_score:
        print("You win!")
        score += 10
    else:
        print("Dealer wins!")
        score -= 10
print("Game over. Thanks for playing!")
cur.execute(
    "INSERT INTO score (USER_ID, GAME_ID, SCORE_VALUE) VALUES (%s, %s, %s)",
    (user_id, 'blackjack', score)
)
con.commit()
print(f"Score saved! Your score: {score}")
input("Press Enter to return to menu...")
```

Arcade Zone Project Documentation

Database Connection (db.py)

```
import mysql.connector as m

con = m.connect(
    host="localhost",
    user="root",
    password="swordsaint",
    database="arcade"
)
cur = con.cursor()
cur.execute("use arcade")
```


Arcade Zone Project Documentation

Sample Outputs and Data

Main Menu Output:

```
=====
MAIN MENU
1. Login
2. Register
3. Exit
=====
```

Registration Output (Sample):

```
REGISTRATION
Enter your username (min 3 chars): johndoe
Enter your email: johndoe@example.com
Enter your mobile number (10 digits): 9876543210
Enter your password: ****
Re-enter your password: ****
Registration successful!
```

Admin Menu Output:

```
ADMIN MENU
1. View all users
2. View all scores
3. View all games
4. Suspend a user
5. Logout
```

View All Users (Sample Data)

ID	Name	Email	Mobile	Password	Role
1	admin	arcadeworld@arcade.com	9999999999	admin@arc	admin
2	johndoe	johndoe@example.com	9876543210	pass1234	player
3	alice	alice@arcade.com	9123456789	alicepass	player
4	bob	bob@arcade.com	9988776655	bobpass	player
5	eve	eve@arcade.com	9001122334	evepass	player

View All Scores (Sample Data)

UserID	Game	Score	Time
--------	------	-------	------

Arcade Zone Project Documentation

2	battleship	5	2025-09-06 12:34:56
2	guess number	10	2025-09-06 12:40:12
3	rock paper scissors	7	2025-09-06 13:10:22
4	rpg	20	2025-09-06 13:15:45
5	blackjack	5	2025-09-06 13:20:10
3	battleship	8	2025-09-06 13:25:33
4	guess number	12	2025-09-06 13:30:01
5	rpg	18	2025-09-06 13:35:27

View All Games

Game ID	Game Name	Description
1	Battleship	Guess the ship's location on a grid.
2	Guess Number	Guess a randomly generated number.
3	Rock Paper Scissors	Classic hand game against computer.
4	RPG Battle Arena	Fight monsters and use potions.
5	BlackJack	Card game against dealer.

Player Menu Output:

```
PLAYER MENU
1. Play Games
2. View Scores
3. Logout
```

Sample Game Output (Battleship):

```
BATTLESHIP
Guess the ship's location (0-9)!
Attempts left 5. Enter your guess: 3
Miss!
Attempts left 4. Enter your guess: 7
Hit! You sunk the battleship!
Score saved! Your score: 5
```

Sample Game Output (Guess the Number):

```
GUESS THE NUMBER
Guess the number between 1 and 100!
Attempts left 7. Enter your guess: 50
Too low!
```

Arcade Zone Project Documentation

```
Attempts left 6. Enter your guess: 75
Too high!
Attempts left 5. Enter your guess: 63
Congratulations! You guessed the number!
Score saved! Your score: 10
```

```
Sample Game Output (Rock Paper Scissors):
ROCK PAPER SCISSORS
Round 1 of 5
Choose rock, paper, or scissors: rock
Computer chose: scissors
You win this round!
...
Game Over! Your score: 7
Score saved!
```

```
Sample Game Output (RPG Battle Arena):
RPG BATTLE ARENA
A wild Goblin appears! Prepare for battle!
Your HP: 30 | Monster HP: 25 | Potions: 3
Choose your action:
1. Attack
2. Heal
3. Run
Enter 1, 2, or 3: 1
You attack the monster for 6 damage!
The monster attacks you for 4 damage!
...
Congratulations! You defeated the monster!
Score saved! Your score: 20
```

```
Sample Game Output (BlackJack):
BLACKJACK
Your cards: [10, 7] (Total: 17)
Dealer's first card: 9
Type 'hit' to get another card, or 'stand' to hold: stand
Dealer's cards: [9, 8] (Total: 17)
It's a draw!
Score saved! Your score: 5
```