PROJECT CONTRIBUTION REPORT

GENPLAY ARCADE

Name: Suraj Kumar K

Date: 05 April 2025

Mentor: Dr. P. Vijayakumari

Problem Statement:

Gen AI Interactive Learning Game

PROBLEM STATEMENT

Traditional education struggles with engagement and personalized learning. This project aims to create an interactive learning game powered by Gen AI to address these issues. The goal is to develop a dynamic tool that:

- Personalizes learning through adaptive content and feedback.
- Increases engagement via gamification and interactive scenarios.
- Facilitates deeper understanding by using Gen AI to generate learning content.

The challenge is to effectively integrate Gen AI into a game, ensuring educational value and ethical considerations. Success would revolutionize learning by creating engaging, personalized educational experiences.

THE PROCESS

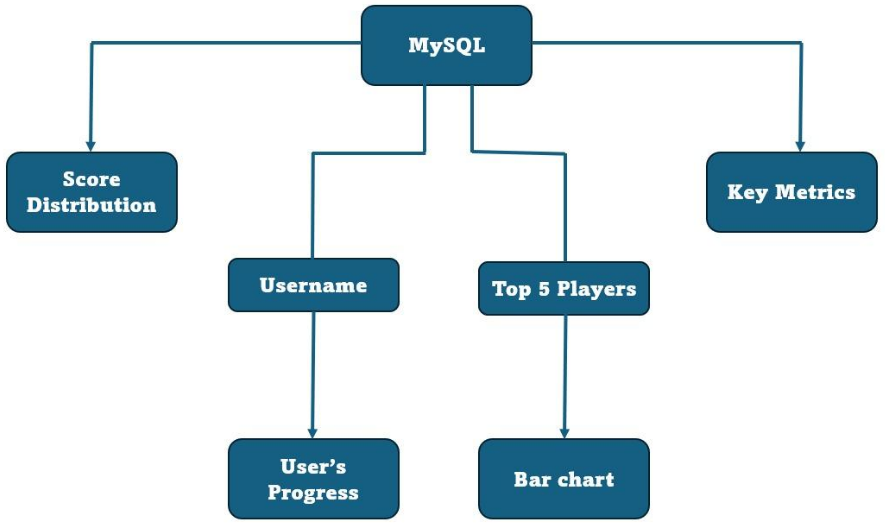
PROJECT DEVELOPMENT AND IMPLEMENTATION

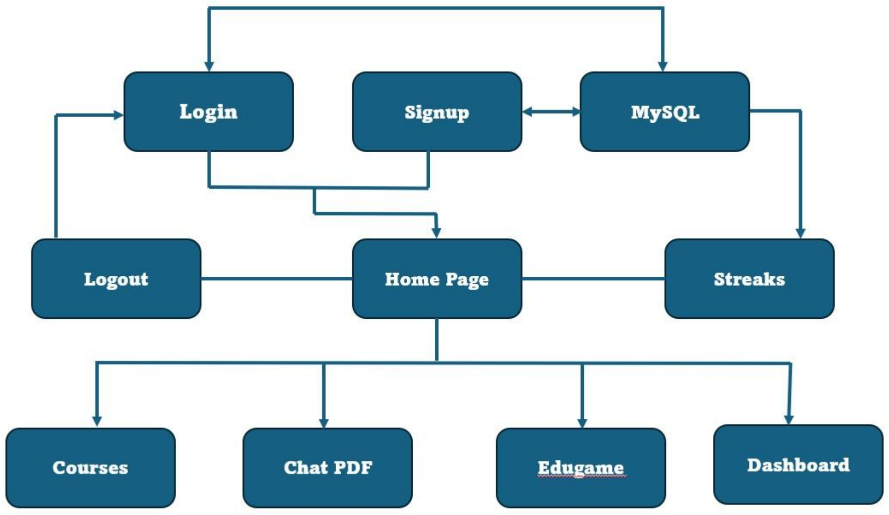
- **User Authentication and Profile Management:** Streamlit was employed to construct robust login and sign-up interfaces. User credentials, along with streak progress, were stored securely within the users table of the MySQL database. To safeguard sensitive user information, hashlib was implemented for password hashing, ensuring that passwords were not stored in plain text. This process involved validating user inputs, handling potential errors, and creating a seamless user experience.

- **Home Page Design and Interactive Features:** The home page, a central hub for the application, was developed using Streamlit. A user-friendly navigation menu was designed to allow easy access to various sections. A visual representation of user activity, the “streak section,” was created using a heatmap. This heatmap, dynamically populated from the users table, provided users with a clear overview of their engagement.

- **Database Management and Data Retrieval:** The MySQL database was central to the application’s functionality. Efficient management of the users and snake_game_scores tables was essential. SQL queries were written to retrieve, insert, update, and delete data, ensuring data integrity and accuracy. Secure database connections were established to protect against unauthorized access.

- **Dashboard Development and Data Visualization:** A comprehensive dashboard page was created within the Streamlit application. This dashboard provided insights into game performance and user activity. Key metrics, such as total players, highest scores, and game counts, were displayed. A leaderboard, sorted by score and date, offered a competitive overview. Bar graphs visualized the top 10 players, while line graphs tracked individual user score progression. A pie chart illustrated the distribution of scores across various ranges. Pandas was utilized for data manipulation and analysis, enabling efficient data processing and visualization. All score data was retrieved from the snake_game_scores table.





SKILL DEVELOPED

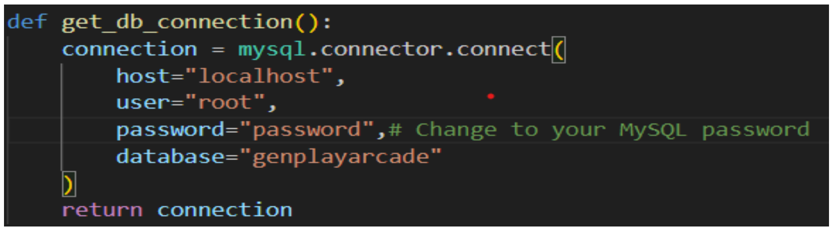
- Streamlit proved to be a powerful tool for rapid web development.
- Skills in managing MySQL databases, specifically the users and snake_game_scores tables, were significantly improved.
- Practical experience in user authentication and password security using hashlib was acquired.
- The ability to create data visualizations within Streamlit, including heatmaps, became a key skill.
- Pandas became a go-to tool for data manipulation and analysis.

PYTHON PACKAGES USED

- Streamlit and Streamlit extras.metric cards
- mysql.connector
- hashlib
- plotly.express
- pandas

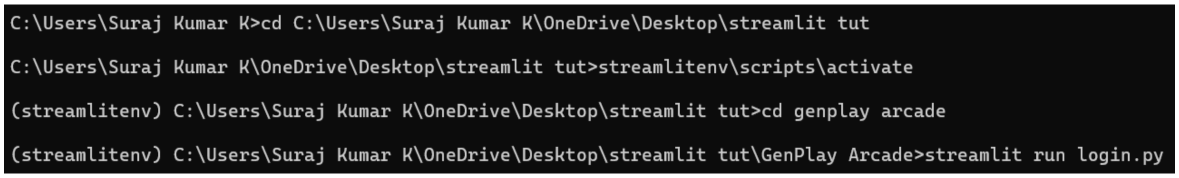
CODE SNIPPETS

To connect to a database:



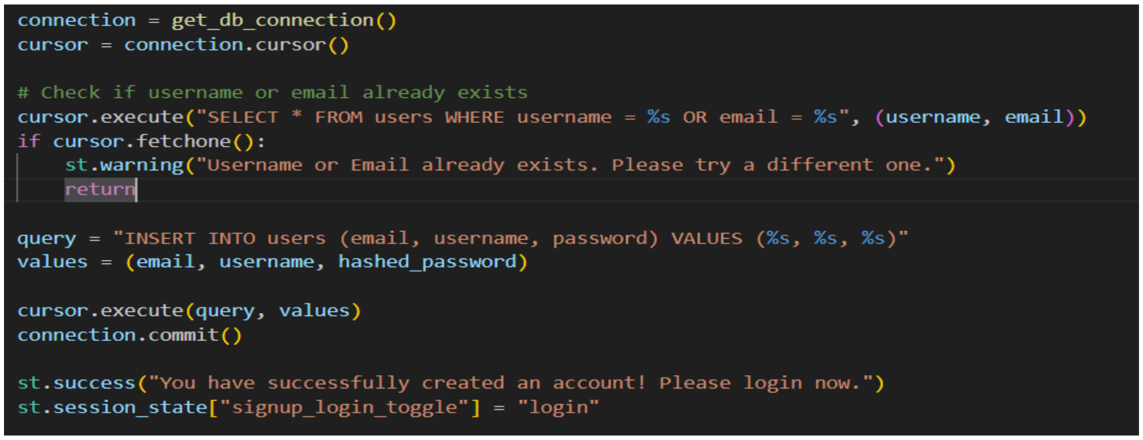
```python
def get_db_connection():
    connection = mysql.connector.connect(
        host="localhost",
        user="root",
        password="password",# Change to your MySQL password
        database="genplayarcade"
    )
    return connection
```

To run the application in command prompt:



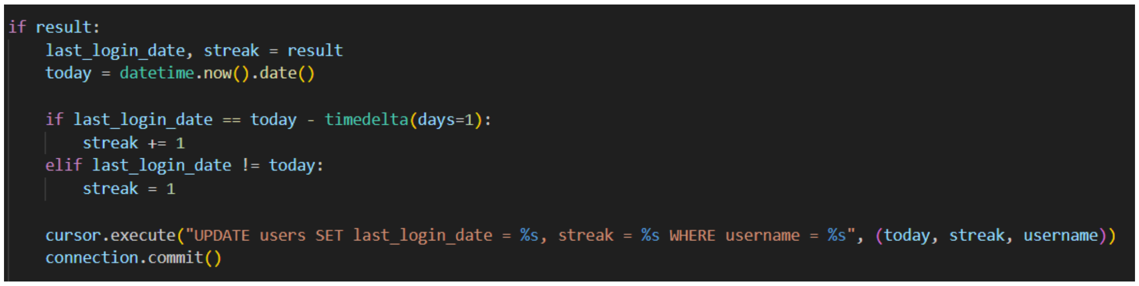
```
C:\Users\Suraj Kumar K>cd C:\Users\Suraj Kumar K\OneDrive\Desktop\streamlit tut

C:\Users\Suraj Kumar K\OneDrive\Desktop\streamlit tut>streamlitenv\scripts\activate

(streamlitenv) C:\Users\Suraj Kumar K\OneDrive\Desktop\streamlit tut>cd genplay arcade

(streamlitenv) C:\Users\Suraj Kumar K\OneDrive\Desktop\streamlit tut\GenPlay Arcade>streamlit run login.py
```

To create sign up:



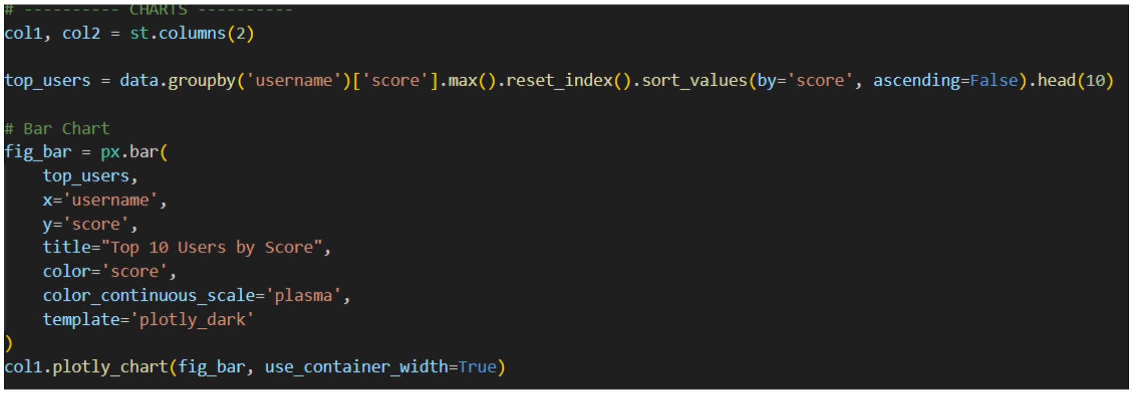
```python
connection = get_db_connection()
cursor = connection.cursor()

# Check if username or email already exists
cursor.execute("SELECT * FROM users WHERE username = %s OR email = %s", (username, email))
if cursor.fetchone():
    st.warning("Username or Email already exists. Please try a different one.")
    return

query = "INSERT INTO users (email, username, password) VALUES (%s, %s, %s)"
values = (email, username, hashed_password)

cursor.execute(query, values)
connection.commit()

st.success("You have successfully created an account! Please login now.")
st.session_state["signup_login_toggle"] = "login"
```

To calculate streak:



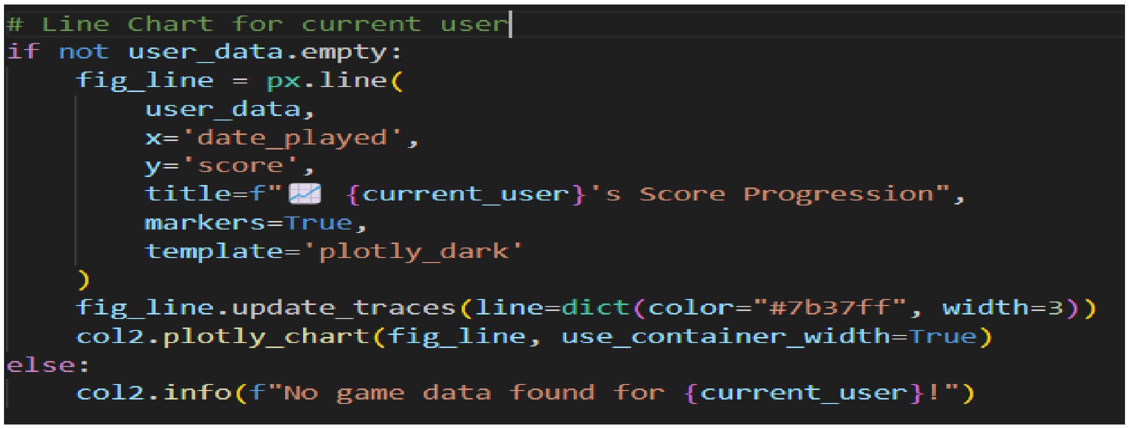
```python
if result:
    last_login_date, streak = result
    today = datetime.now().date()

    if last_login_date == today - timedelta(days=1):
        streak += 1
    elif last_login_date != today:
        streak = 1

    cursor.execute("UPDATE users SET last_login_date = %s, streak = %s WHERE username = %s", (today, streak, username))
    connection.commit()
```

To create a bar chart of top 10 scores:



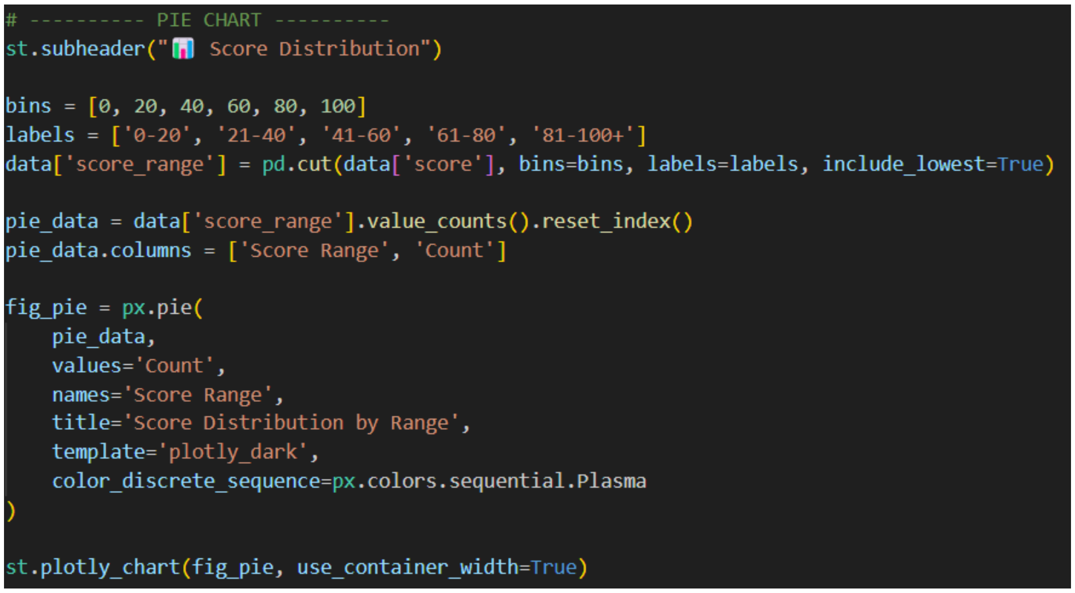
```
# ---------- CHARTS ----------
col1, col2 = st.columns(2)

top_users = data.groupby('username')['score'].max().reset_index().sort_values(by='score', ascending=False).head(10)

# Bar Chart
fig_bar = px.bar(
    top_users,
    x='username',
    y='score',
    title="Top 10 Users by Score",
    color='score',
    color_continuous_scale='plasma',
    template='plotly_dark'
)
col1.plotly_chart(fig_bar, use_container_width=True)
```

To create a line chart of user's progress:



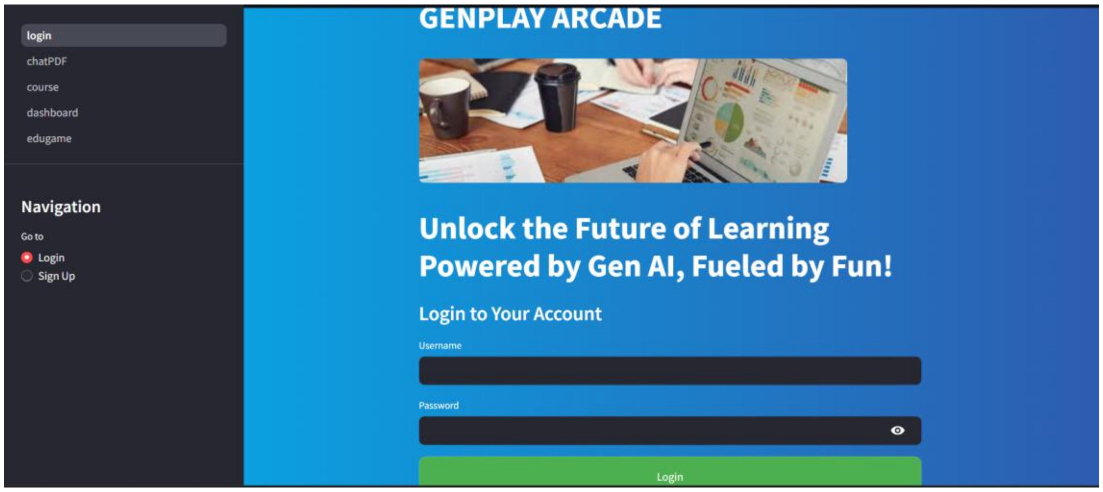
```
# Line Chart for current user
if not user_data.empty:
    fig_line = px.line(
        user_data,
        x='date_played',
        y='score',
        title=f"📈 {current_user}'s Score Progression",
        markers=True,
        template='plotly_dark'
    )
    fig_line.update_traces(line=dict(color="#7b37ff", width=3))
    col2.plotly_chart(fig_line, use_container_width=True)
else:
    col2.info(f"No game data found for {current_user}!")
```

To create a pie chart for score distribution:



```
# ---------- PIE CHART ----------
st.subheader("📊 Score Distribution")

bins = [0, 20, 40, 60, 80, 100]
labels = ['0-20', '21-40', '41-60', '61-80', '81-100+']
data['score_range'] = pd.cut(data['score'], bins=bins, labels=labels, include_lowest=True)

pie_data = data['score_range'].value_counts().reset_index()
pie_data.columns = ['Score Range', 'Count']

fig_pie = px.pie(
    pie_data,
    values='Count',
    names='Score Range',
    title='Score Distribution by Range',
    template='plotly_dark',
    color_discrete_sequence=px.colors.sequential.Plasma
)

st.plotly_chart(fig_pie, use_container_width=True)
```

OUTPUT



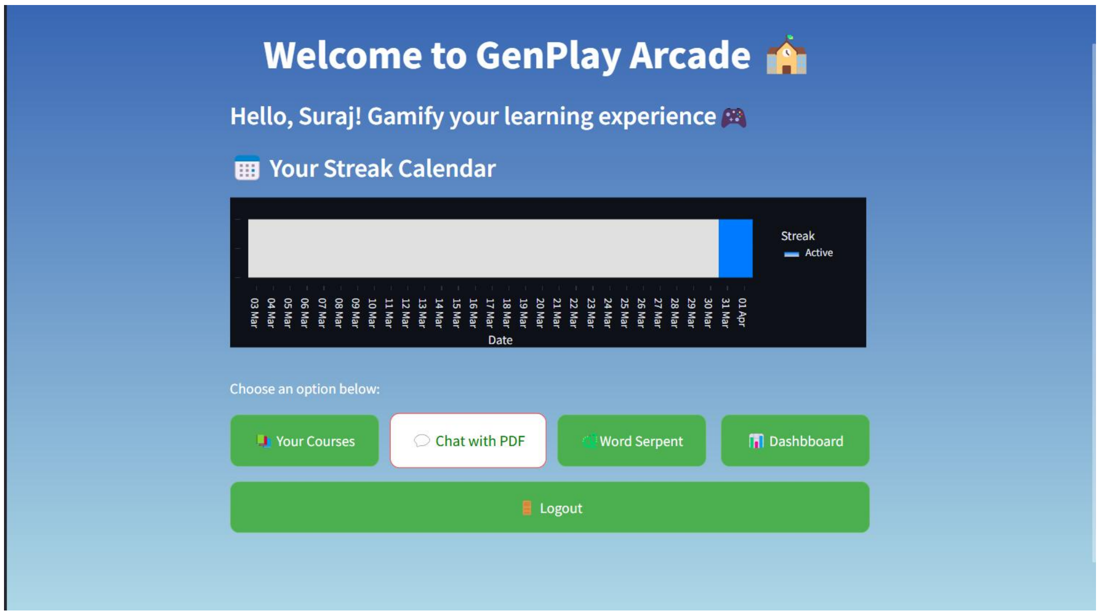
GENPLAY ARCADE

Unlock the Future of Learning Powered by Gen AI, Fueled by Fun!

Login to Your Account

Username:

Password:

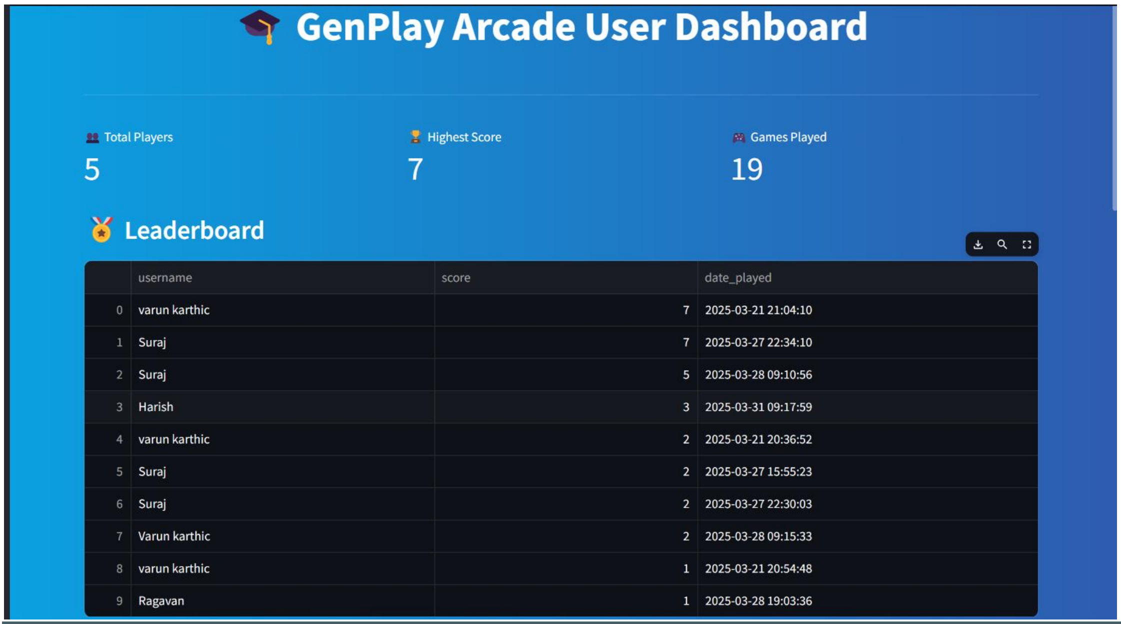


Welcome to GenPlay Arcade 🏫

Hello, Suraj! Gamify your learning experience 🎮

Your Streak Calendar

Choose an option below:

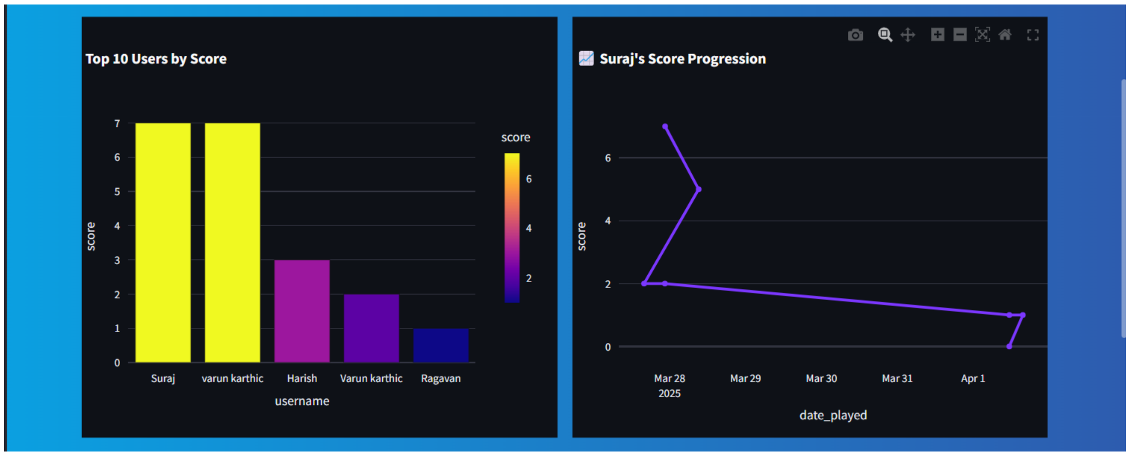


GenPlay Arcade User Dashboard

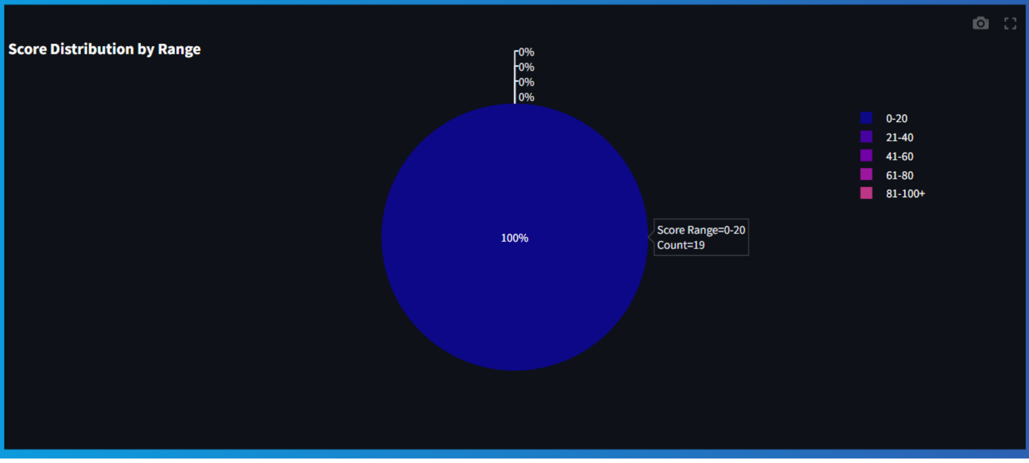
Total Players	Highest Score	Games Played
5	7	19

Leaderboard

	username	score	date_played
0	varun karthic	7	2025-03-21 21:04:10
1	Suraj	7	2025-03-27 22:34:10
2	Suraj	5	2025-03-28 09:10:56
3	Harish	3	2025-03-31 09:17:59
4	varun karthic	2	2025-03-21 20:36:52
5	Suraj	2	2025-03-27 15:55:23
6	Suraj	2	2025-03-27 22:30:03
7	Varun karthic	2	2025-03-28 09:15:33
8	varun karthic	1	2025-03-21 20:54:48
9	Ragavan	1	2025-03-28 19:03:36



Score Distribution


Score Distribution by Range
Score Range=0-20
Count=19