Technical Documentation

1. Project Information

- **Project Title** Data analysis for Mental Health & Technology Usage
- **Course/Track:** Data analysis specialist
- **Team Members:**

    1. Nouran Adel Younes
    2. Ahmed Wagdy Swilam
    3. Mariem Mohamed Ebrahim
    4. Hager Saleh Mahmoud
    5. Mayar Kamal Maken

2. Objectives

The analysis examines the correlation between technology usage, social media interaction, and gaming hours with mental health indicators like stress levels, sleep hours, and overall well-being.

Data sources:

- Mental Health & Technology Usage Dataset from Kaggle website.
- Google Forms survey data on technology usage, social media engagement, gaming hours, stress levels, sleep patterns, and overall well-being.

3. Data collection process

We collect data by conducting a survey using Google Forms along with a relevant Mental Health & Technology Usage Dataset from Kaggle website, which is separated into two CSV file to investigate the link between technology usage, social media engagement, gaming hours, stress levels, sleep patterns and gain a comprehensive understanding of the relationship between technology and mental health.

Name	User_I	Age	Gende	Technology_Usage_Hou	Social_Media_Usage_Hou	Gaming_Hours	Screen_Time_Hours	Mental_Health_Status	Stress_Level	Sleep_Hours	Physical_Activity_Hours	Support_Systems_Acces	Work_Environment_Impa	Online_Support_
Nourhan	USER-100	21	Famale	4-6 Hours	2-4 Hours	1-2 Hours		Good	Medium	8-10 Hours	2-3 Hours	Yes	Neutral	Yes
هاجر مصط	USER-100	20	Famale	8-10 Hours	4-6 Hours	2-4 Hours	10	Good	Medium	8-10 Hours	2-3 Hours	No	Negative	No
Lara Ahme	USER-100	20	Female	4-6 Hours	2-4 Hours	2-4 Hours	5	Good	Medium	4-6 Hours	5-6 Hours	Yes	Neutral	Yes
	USER-100	21	Hghj	4-6 Hours	2-4 Hours	2-4 Hours		Poor	High	6-8 Hours	3-4 Hours	No	Positive	No
منه الله احمد	USER-100	20	Egyptian	8-10 Hours	2-4 Hours	1-2 Hours	6	Excellent	Medium	6-8 Hours	3-4 Hours	Yes	Positive	Yes
Mennatull	USER-100	22	Female	6-8 Hours	2-4 Hours	1-2 Hours	4	Fair	Medium	8-10 Hours	2-3 Hours		Neutral	
منه صالح	USER-100	18	انثي	2-4 Hours	4-6 Hours	1-2 Hours	5	Poor	Medium	6-8 Hours	2-3 Hours	No	Neutral	No
Yasmine h	USER-100	18	Female	6-8 Hours	4-6 Hours	1-2 Hours	8	Poor	Medium	6-8 Hours	2-3 Hours	No	Neutral	Yes
ياسمين	USER-100	18	انثى	6-8 Hours	4-6 Hours	1-2 Hours	6	Good	Medium	6-8 Hours	2-3 Hours	Yes	Neutral	Yes
Sondos	USER-100	18	Female	6-8 Hours	2-4 Hours	1-2 Hours	10	Good	Medium	8-10 Hours	2-3 Hours	Yes	Positive	Yes
ساره السيد	USER-100	18	Female	8-10 Hours	4-6 Hours	4-6 Hours	8	Good	Medium	8-10 Hours	2-3 Hours	Yes	Neutral	
Arwa Fath	USER-100	18	Female	2-4 Hours	4-6 Hours	1-2 Hours	4-6 Hours	Good	High	6-8 Hours	2-3 Hours	Yes	Positive	Yes
Yasmin N	USER-100	18	Female	2-4 Hours	4-6 Hours	1-2 Hours	12	Fair	High	8-10 Hours	2-3 Hours		Neutral	
Nourhan	USER-100	20	feminine	2-4 Hours	4-6 Hours	1-2 Hours	6	Good	Medium	4-6 Hours	5-6 Hours	Yes	Positive	Yes

Survey responses as excel file

4. Data exploration and cleaning

By utilizing Google Colab and Python, we efficiently explored and cleaned the dataset. We addressed issues such as missing values, duplicates, and outliers using Python's powerful libraries like Pandas. This allowed us to perform data inspection, apply transformations, and ensure the dataset was properly prepared for further analysis, all within a collaborative environment.

Data exploration and cleaning for the survey response sheet

Google Colab Notebook : link to the notebook

➢ Data exploration

- Importing Required Libraries

```
import pandas as pd
import seaborn as sns
```

- Loading the Dataset
    - The dataset is loaded into a Pandas DataFrame for easier manipulation.

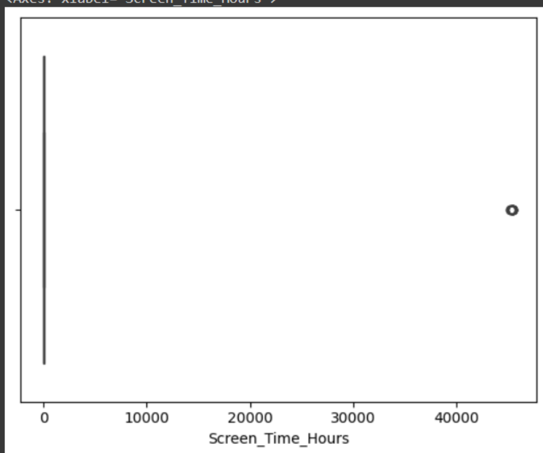
```
#load data set
df = pd.read_csv('/content/Untitled form (Responses).csv')

#Qick view of data
df.head(3)
df.shape
```

- Check data problems

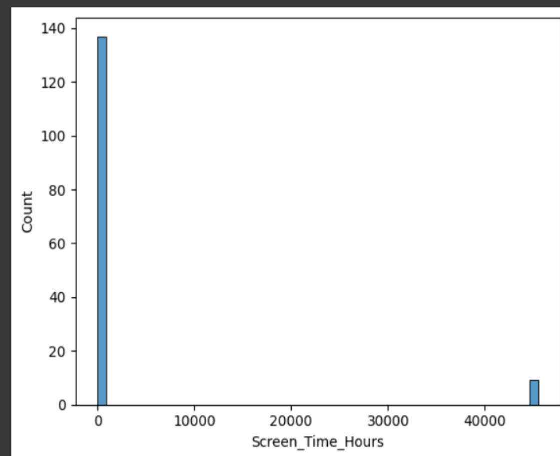
```
[16] df.info()                  # the last 3 coulmns are emity need to be dropped

     df.dtypes                  # change data type from object to float:
                                       # 'age' , 'Technology_Usage_Hours'
                                       #'Social_Media_Usage_Hours','Gaming_Hours'
                                       # 'Screen_Time_Hours', 'Sleep_Hours'
                                       #'Physical_Activity_Hours'

[18] df.isnull().sum()                  # 3 coulmns have nulls

     df.duplicated().sum()              # 2 duplicate need to be dropped
```

- Chek for outlier
  - Using Seaborn Library, we plot a histogram and boxplot to check if our data have outlier .

```
sns.boxplot(x=df['Screen_Time_Hours'])        #From the boxplot there is outli
<Axes: xlabel='Screen_Time_Hours'>
```




```
sns.histplot(df['Screen_Time_Hours'],bins=50)
plt.show()
```



➢ Data cleaning

- Handling data type problems
  - In the ***'Age'*** column some of the values are string so we replace them with numeric value

```
df['Age'].unique()                    # '22!' and '21years ' must be replaced with numeric value

array(['21', '20', '22', '18', '16', '19', '22!', '23', '24', '21years ',
       '26', '25', '32', '36', '38', '27', '30', '15', '17', '14', '43',
       '60'], dtype=object)
```

```
[ ] df['Age'].replace({'21years ': 21, '22!': 22} , inplace=True)
    df['Age'].unique()                #all values become numeric

array(['21', '20', '22', '18', '16', '19', 22, '23', '24', 21, '26', '25',
       '32', '36', '38', '27', '30', '15', '17', '14', '43', '60'],
      dtype=object)
```

  - In the ***'Gender'*** column some of the entered values are written in Arabic and some are numeric, so we replace these values become [Male , Female]

```
[ ] df['Gender'].unique()

array(['Famale', 'Female ', 'Hghj', 'Egyptian ', 'انثى' ,' انثي',
       'Female', 'feminine', 'Femail', 'F', 'female', 'انثي', 'Male',
       'male', 'male ', '18' ,'ذكر', 'Male ', 'امرأه ', 'Boy', 'Girl',
       'دكر', 'female ', 'Girl ', 'feminine ', '43' ,' ذكر' ,'انثيبي',
       'Women', '  Female', 'Maale', 'Mail', 'مصري ',
       'Abdelrahmantolba962@gmail.com '], dtype=object)
```

```
[ ] # Replace specific values in the 'Gender' column
    df['Gender'] = df['Gender'].replace({
        'انثى': 'Female',
        'F': 'Female',
        'Egyptian': 'Female',
        'Female ': 'Female',  # To handle any extra spaces
        'Male ': 'Male',      # To handle any extra spaces
        'Hghj': 'Male', # Example to replace invalid entries
        'ذكر': 'Male',          # Arabic for Male

    })

    # Ensure only 'Male' or 'Female' are in the Gender column
    df['Gender'] = df['Gender'].apply(lambda x: 'Male' if x in ['Male', 'ذكر'] else 'Female' if x in ['Female', 'انثى'] else 'Other')
    df['Gender'].unique()

array(['Other', 'Female', 'Male'], dtype=object)
```

- Change data type of numeric column to float and integer

```
df.columns                    # know the columns name to change its data type
```

```
Index([' Name', 'User_ID', 'Age', 'Gender', 'Technology_Usage_Hours',
       'Social_Media_Usage_Hours', 'Gaming_Hours', 'Screen_Time_Hours',
       'Mental_Health_Status', 'Stress_Level', 'Sleep_Hours',
       'Physical_Activity_Hours', 'Support_Systems_Access',
       'Work_Environment_Impact', 'Online_Support_Usage'],
      dtype='object')
```

```
#change data type
df['Age'] = df['Age'].astype(int)
df['Gaming_Hours'] = df['Gaming_Hours'].astype(float)
df['Sleep_Hours'] = df['Sleep_Hours'].astype(float)
df['Physical_Activity_Hours'] = df['Physical_Activity_Hours'].astype(float)
df['Screen_Time_Hours'] = df['Screen_Time_Hours'].astype(float)
df['Technology_Usage_Hours'] = df['Technology_Usage_Hours'].astype(float)
df['Social_Media_Usage_Hours'] = df['Social_Media_Usage_Hours'].astype(float)
df.dtypes
```

- Handling missing data
  - There are 3 empty columns that need to be dropped

```
df = df.drop(['Unnamed: 15', 'Unnamed: 16','Unnamed: 17','@'], axis=1)
df
```

  - Handling null values by replacing it with mean for the columns that have numeric value and with mode with the column that has categorical value
    - **Numeric value**

```
# Use .any() to check if any value in the column is null
if df['Screen_Time_Hours'].isnull().any():
    # Calculate the sum of the three columns
    df['Screen_Time_Hours'] = df['Screen_Time_Hours'].fillna(df['Technology_Usage_Hours'] + df['Social_Media_Usage_Hours'] + df['Gaming_Hours'])
df['Screen_Time_Hours']
```

    - **Categorical value**

```
#mode of the Support_Systems_Access
mode_Support_Systems_Access = df['Support_Systems_Access'].mode()[0]
mode_Support_Systems_Access
```

```
'Yes'
```

```
#Fill null value of the Support_Systems_Access with the mode
df['Support_Systems_Access'] = df['Support_Systems_Access'].fillna(mode_Support_Systems_Access)
```

```
[25] #mode of the Support_Systems_Access
df['Online_Support_Usage'].mode()[0]
```

```
'Yes'
```

```
# Fill empty cells in the 'Online Support Usage' column with 'Yes'
df['Online_Support_Usage'].fillna('Yes', inplace=True)
```

- Removing duplicates
    - Duplicates can significantly impact the quality, accuracy, and reliability of your data, and lead to inaccurate results in the analysis

```
[30] df = df.drop_duplicates()
```

```
df.duplicated().sum()
```

```
0
```

- Handling outlier

```
[ ] Q1 = df['Screen_Time_Hours'].quantile(0.25)
    Q3 = df['Screen_Time_Hours'].quantile(0.75)
    IQR = Q3 - Q1
    lower_limit = Q1 - 1.5 * IQR
    upper_limit = Q3 + 1.5 * IQR
```

```
print(IQR)
print(lower_limit)
print(upper_limit)
```

```
5.75
-2.625
20.375
```

- We solve the outlier problem by replacing the values that are greater than the upper limit with the median

```
# Calculate the median of 'Screen_Time_Hours'
median_Screen_Time = df['Screen_Time_Hours'].median()

# Replace values greater than the upper limit with the median
df.loc[df['Screen_Time_Hours'] > 19.125, 'Screen_Time_Hours'] = median_Screen_Time
```

Data exploration and cleaning for Mental Health & Technology Usage Dataset from Kaggle website

Google Colab Notebook : [link to the notebook]

➢ Data exploration

- Importing Required Libraries and load data

```
[ ] import pandas as pd
```

```
[ ] df = pd.read_csv('/content/Unclean_first_dataset.csv')
    df2 = pd.read_csv('/content/second_dataset.csv')
```

- Check data problems

    - In **'df'** ,there are 3 duplicates need to be dropped and 6 columns have null value

```
[ ] df.duplicated().sum()
3
```

```
df.isnull().sum()
```

	0
User_ID	0
Technology_Usage_Hours	4
Social_Media_Usage_Hours	3
Gaming_Hours	5
Screen_Time_Hours	3
Sleep_Hours	4
Physical_Activity_Hours	4
Work_Environment_Impact	0

dtype: int64

df

```
[ ] df2.duplicated().sum()
3
```

```
df2.isnull().sum()
```

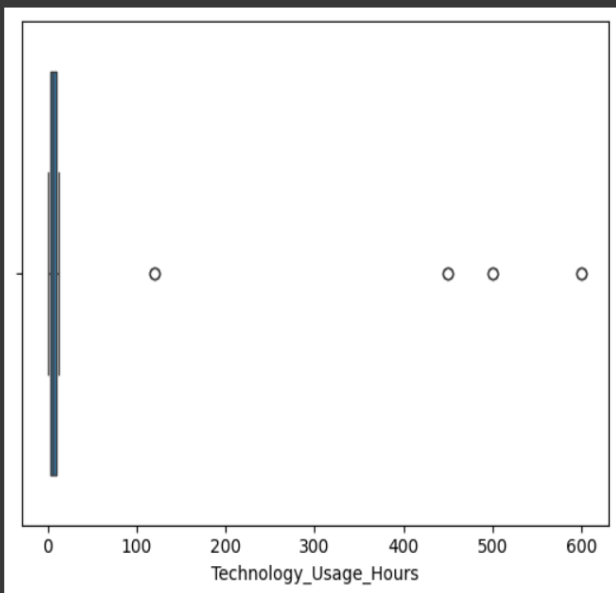
	0
User_ID	0
Age	3
Gender	3
Mental_Health_Status	1
Stress_Level	3
Support_Systems_Access	3
Online_Support_Usage	4

dtype: int64

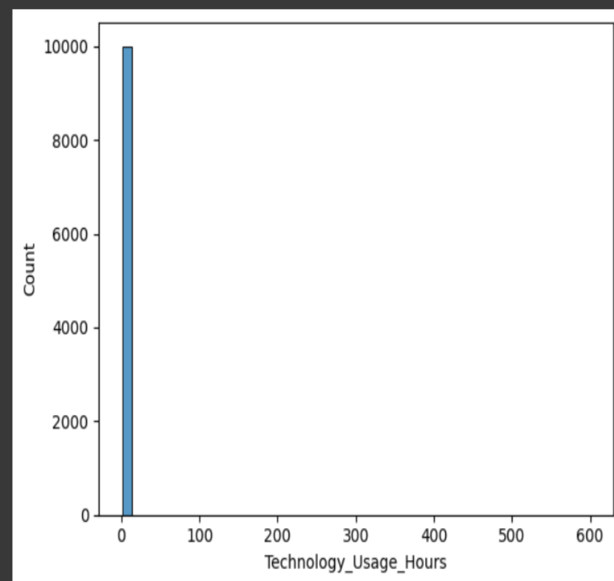
df2

- Checking for outliers
    - By ploting a histogram and boxplot for first dataset we found outliers that we need to handle them in **‘Technology_Usage_Hours' column and ‘Screen_Time_Hours’ column**

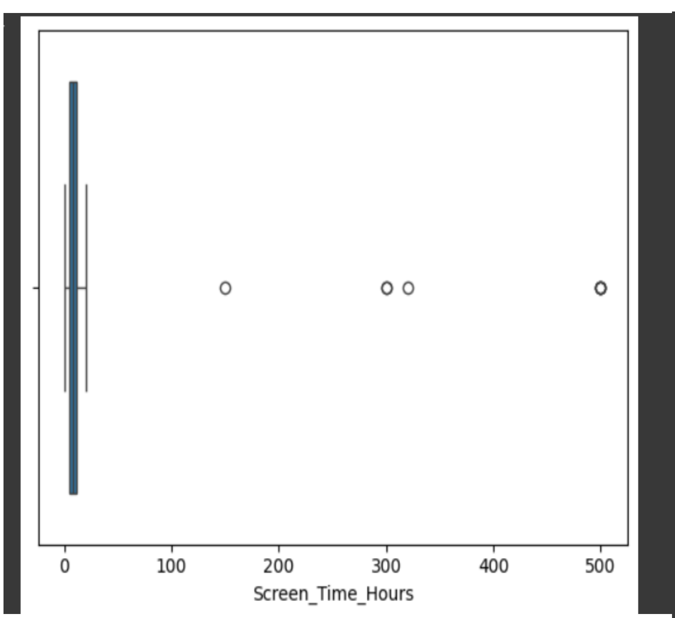
```
sns.boxplot(x=df['Technology_Usage_Hours'])
plt.show()
```

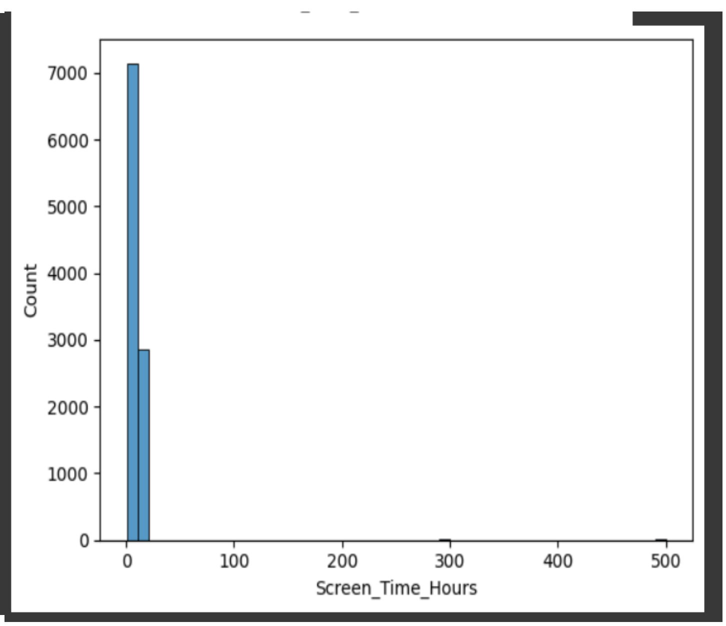



```
sns.histplot(df['Technology_Usage_Hours'],bins=50)
plt.show()
```




```
sns.boxplot(x=df['Screen_Time_Hours'])
plt.show()
sns.histplot(df['Screen_Time_Hours'],bins=50)
plt.show()
```





➢Data cleaning

- Handling outliers
  - We calculate the IQR for the ***'Technology_Usage_Hours'*** column and ***'Screen_Time_Hours'*** column respectively. Then we determine the lower and upper limits for identifying potential outliers. The lower limit is set to 1.5 times the IQR below the first quartile (Q1), and the upper limit is 1.5 times the IQR above the third quartile (Q3). Any data points outside these limits can be considered outliers.

```
Q1=df['Technology_Usage_Hours'].quantile(0.25)
Q3=df['Technology_Usage_Hours'].quantile(0.75)
IQR=Q3-Q1
lower_limit=Q1-1.5*IQR
upper_limit=Q3+1.5*IQR
```

```
Q1_Screen_time=df['Screen_Time_Hours'].quantile(0.25)
Q3_Screen_time=df['Screen_Time_Hours'].quantile(0.75)
IQR_Screen_time=Q3-Q1
lower_limit_Screen_time=Q1_Screen_time-1.5*IQR
upper_limit_Screen_time=Q3_Screen_time+1.5*IQR
```

  - We replace values greater than upper limit in the ***'Technology_Usage_Hours'*** column with the column's median, using a lambda function. We do the same thing to ***'Screen_Time_Hours'*** column

```
median_value = df['Technology_Usage_Hours'].median()

df['Technology_Usage_Hours'] = df['Technology_Usage_Hours'].apply(lambda x: median_value if x > 17.41 else x)
```

```
[ ] median_value = df['Screen_Time_Hours'].median()

    df['Screen_Time_Hours'] = df['Screen_Time_Hours'].apply(lambda x: median_value if x > 17.41 else x)
```

- Handling missing data

  - **Numeric values**

    - We fill missing values (NaN) in several columns with their respective column means.

```
df['Technology_Usage_Hours'].fillna(df['Technology_Usage_Hours'].mean(),inplace=True)
df['Social_Media_Usage_Hours'].fillna(df['Social_Media_Usage_Hours'].mean(),inplace=True)
df['Gaming_Hours'].fillna(df['Gaming_Hours'].mean(),inplace=True)
df['Screen_Time_Hours'].fillna(df['Screen_Time_Hours'].mean(),inplace=True)
df['Sleep_Hours'].fillna(df['Sleep_Hours'].mean(),inplace=True)
df['Physical_Activity_Hours'].fillna(df['Physical_Activity_Hours'].mean(),inplace=True)
```

- We make a model to fill missing values in the ***'Age'*** column of **df2** by employing a decision tree regressor. It utilizes the following categorical features for prediction: 'Gender'**, 'Mental_Health_Status', 'Stress_Level', 'Support_Systems_Access', and 'Online_Support_Usage'**. The process involves one-hot encoding these features, training the model on records with known 'Age' values, and then predicting and filling the missing 'Age' entries.

```
features = ['Gender', 'Mental_Health_Status', 'Stress_Level', 'Support_Systems_Access', 'Online_Support_Usage']

df2_encoded = pd.get_dummies(df2[features])

known_age = df2[df2['Age'].notnull()]
unknown_age = df2[df2['Age'].isnull()]

X_train = pd.get_dummies(known_age[features])
y_train = known_age['Age']

X_test = pd.get_dummies(unknown_age[features])
X_test = X_test.reindex(columns=X_train.columns, fill_value=0)
```

```
from sklearn.tree import DecisionTreeRegressor

model = DecisionTreeRegressor()

model.fit(X_train, y_train)

predicted_ages = model.predict(X_test)

predicted_ages = predicted_ages.round().astype(int)

df2.loc[df2['Age'].isnull(), 'Age'] = predicted_ages
```

- **categorical values**
    - We make a code to iterate through all columns in df2. If a column has a data type of 'object', it fills missing values (NaN) with mode of that column, updating the column in place.

```python
for col in df2.columns:
    if df2[col].dtype == 'object':
     df2[col].fillna(df2[col].mode()[0], inplace=True)
```

- Removing duplicates
    - We drop duplicates from both **df** and **df2**

```python
df.drop_duplicates(inplace=True)
```

```python
df.duplicated().sum()
```

```
0
```

```python
df2.drop_duplicates(inplace=True)
```

```python
df2.duplicated().sum()
```

```
0
```

5. Data merging

We used SQL to join different datasets together. By using SQL queries, we combined tables based on shared columns, creating a complete dataset for analysis. This helped us easily merge data from different sources and made the process more efficient for our analysis.

First : we import the 3 data sets into SQL server management studio

Second : we join the first and second date sets together

- The SQL query combines data from two tables into a single table by joining them based on matching user IDs. The combined data is then sorted by user ID and optionally stored in a new table(first_second_data_table).

SQLQuery1.sql - D...\MR.SWAILAM (53))*

```sql
SELECT
    t1.User_ID AS User_ID,
    t1.Technology_Usage_Hours,
    t1.Social_Media_Usage_Hours,
    t1.Gaming_Hours,
    t1.Screen_Time_Hours,
    t1.Sleep_Hours,
    t1.Physical_Activity_Hours,
    t1.Work_Environment_Impact,
    t2.Age,
    t2.Gender,
    t2.Mental_Health_Status,
    t2.Stress_level,
    t2.Support_Systems_Access,
    t2.Online_Support_Usage
INTO first_second_data_Table
FROM
    first_data t1
JOIN
    second_data t2
ON
    t1.User_ID = t2.User_ID
Order by
    t1.User_ID
```

100 %

Results | Messages

	User_ID	Technology_Usage_Hours	Social_Media_Usage_Hours	Gaming_Hours	Screen_Time_Hours	Sleep_Hours	Physical_Activity_Hours	Work_Environment_Impact	Age	Gender	Mental_Health_S
1	USER-00001	6.57	6	0.68	12.36	8.01	6.71	Negative	23	Female	Good
2	USER-00002	3.01	2.57	3.74	7.61	7.28	5.88	Positive	21	Male	Poor
3	USER-00003	3.04	6.14	1.26	3.16	8.04	9.81	Negative	51	Male	Fair
4	USER-00004	3.84	4.48	2.59	13.08	5.62	5.28	Negative	25	Female	Excellent

Third : we concatenate the survey date set with the (first_second_data_table)

- The SQL query combines data from two tables into a single table by first inserting data from the first table and then updating the combined table with data from the second table for users who are not already included. The resulting table can be used for further data analysis and insights into user behavior patterns.

SQLQuery1.sql - D...\MR.SWAILAM (53))*

```sql
INSERT INTO first_second_data3_Table (
    User_ID,
    Technology_Usage_Hours,
    Social_Media_Usage_Hours,
    Gaming_Hours,
    Screen_Time_Hours,
    Sleep_Hours,
    Physical_Activity_Hours,
    Work_Environment_Impact,
    Age,
    Gender,
    Mental_Health_Status,
    Stress_Level,
    Support_Systems_Access,
    Online_Support_Usage
)
SELECT
    User_ID,
    Technology_Usage_Hours,
    Social_Media_Usage_Hours,
    Gaming_Hours,
    Screen_Time_Hours,
    Sleep_Hours,
    Physical_Activity_Hours,
    Work_Environment_Impact,
    Age,
    Gender,
    Mental_Health_Status,
    Stress_Level,
    Support_Systems_Access,
    Online_Support_Usage
FROM
    response_data
WHERE
    User_ID NOT IN (SELECT User_ID FROM first_second_data3_Table);
```

Messages

(146 rows affected)

Completion time: [illegible]

6. Data analysis

- Once the data was cleaned and combined, we formulated several key questions to extract insights using SQL queries. We aimed to identify the most valuable insights in the dataset and check if the data centered around specific themes. We examined how many individuals fell into different mental health status categories and analyzed the age distribution within those categories. We also explored factors impacting mental health, such as stress levels and work environment, and assessed whether support systems could help improve the relationship between stress and mental health. Additionally, we investigated the influence of sleep hours on mental health and how physical activity, screen time, and gaming affect stress levels. This process allowed us to better understand the various factors influencing mental health and stress.

1- What is the number of people in the mental health status categories?

SQLQuery1.sql -...MARAM\Maram (65))*

```sql
SELECT mental_health_status, COUNT(*) AS number_of_people
FROM Final_Data
GROUP BY mental_health_status;
```

	mental_health_status	number_of_people
1	Good	1872
2	Poor	1780
3	Excellent	2579
4	Fair	3931

2- What is the number of each age interval in mental health status categories?

```sql
SELECT age_interval, mental_health_status, COUNT(*) AS number_of_people
FROM (
    SELECT CASE
        WHEN age < 20 THEN 'Under 20'
        WHEN age BETWEEN 20 AND 29 THEN '20-29'
        WHEN age BETWEEN 30 AND 39 THEN '30-39'
        WHEN age BETWEEN 40 AND 49 THEN '40-49'
        ELSE '50 and above'
    END AS age_interval,
    mental_health_status
    FROM Final_Data
) AS age_groups
GROUP BY age_interval, mental_health_status
ORDER BY age_interval;
```

	age_interval	mental_health_status	number_of_people
3	20-29	Excellent	561
4	20-29	Poor	378
5	30-39	Excellent	548
6	30-39	Good	369
7	30-39	Fair	768
8	30-39	Poor	382
9	40-49	Fair	827
10	40-49	Excellent	496
11	40-49	Good	376
12	40-49	Poor	368
13	50 and above	Excellent	869
14	50 and above	Fair	1305
15	50 and above	Good	608
16	50 and above	Poor	576
17	Under 20	Good	75
18	Under 20	Poor	76
19	Under 20	Excellent	105
20	Under 20	Fair	173

3- How do stress levels impact mental health status?

```sql
SELECT mental_health_status,
        AVG(
            CASE
                WHEN stress_level = 'low' THEN 1
                WHEN stress_level = 'medium' THEN 2
                WHEN stress_level = 'high' THEN 3
            END
        ) AS average_stress
FROM Final_Data
GROUP BY mental_health_status;
```

	mental_health_status	average_stress
1	Good	1
2	Poor	2
3	Excellent	1
4	Fair	2

4- How does the work environment impact mental health status?

```sql
SELECT Work_Environment_Impact, COUNT(*) AS number_of_people
FROM Final_Data
WHERE mental_health_status IS NOT NULL
GROUP BY Work_Environment_Impact;
```

	Work_Environment_Impact	number_of_people
1	Neutral	3401
2	Positive	3411
3	Negative	3350

5- What is the most useful insight in data ?

6- Is the data revolving around specific something ?

7- Is support can resolve the problem in relationship between Stress, Work impact on mental health status?

8- Which things impact stress level?

9- **Do sleep hours affect mental health status ?**

10- **How do sleep hours affect directly and not directly on mental health status?**

11- **How do the average hours spent like physical activity, screen time and gaming hours impact on stress levels?**

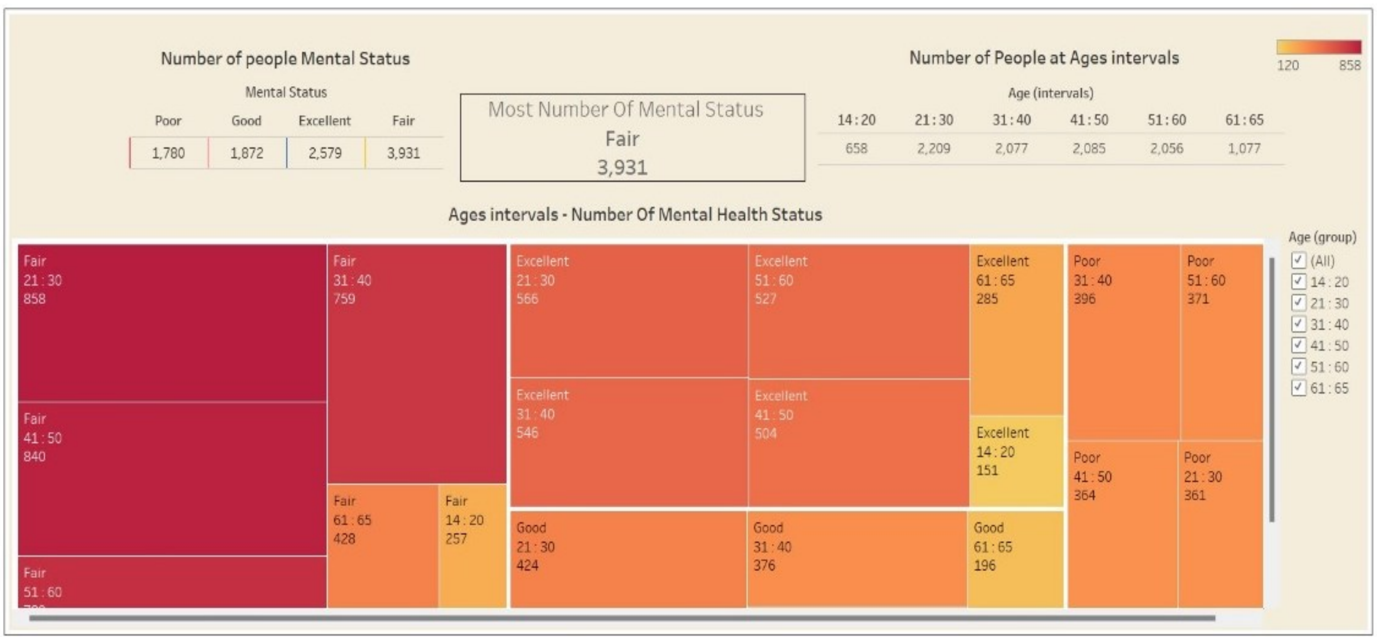
12- **How do work environment impact on stress level?**

7. Data visualization

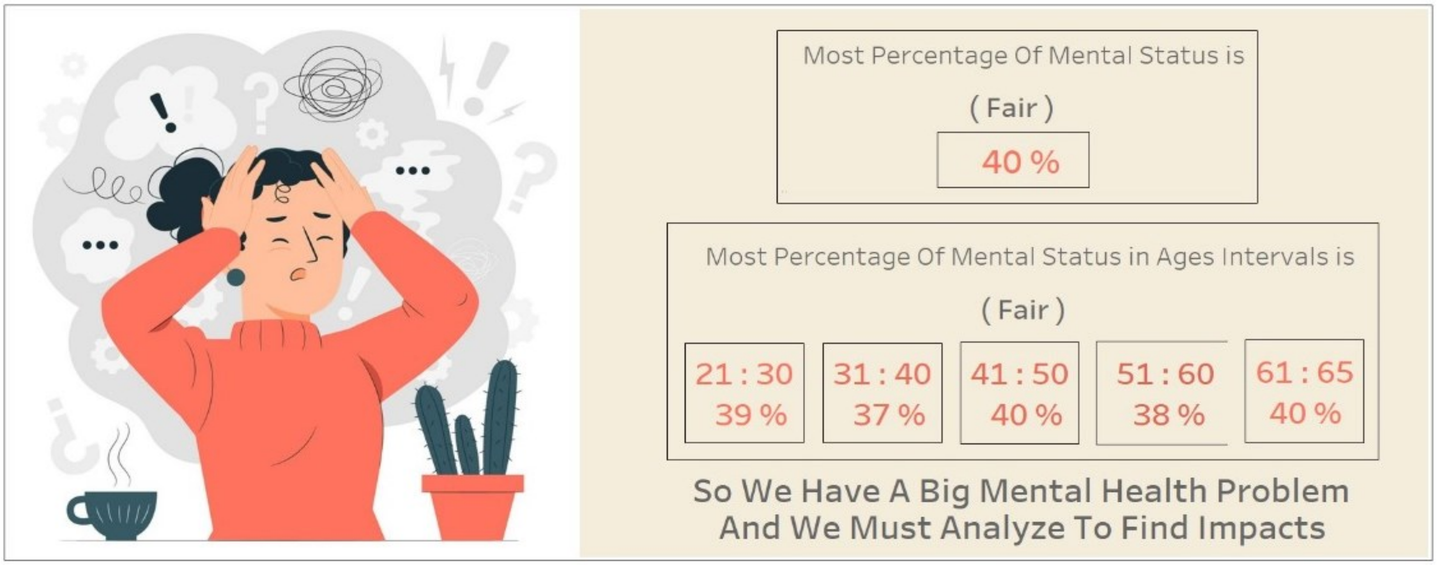
- We used Tableau to create an interactive and insightful dashboard that helps visualize the key findings from our data analysis. Tableau's powerful data visualization tools allowed us to present complex insights in a more accessible and user-friendly way. By transforming raw data into visual formats such as charts and graphs.

- **Link to the dashboards : dashboard link**

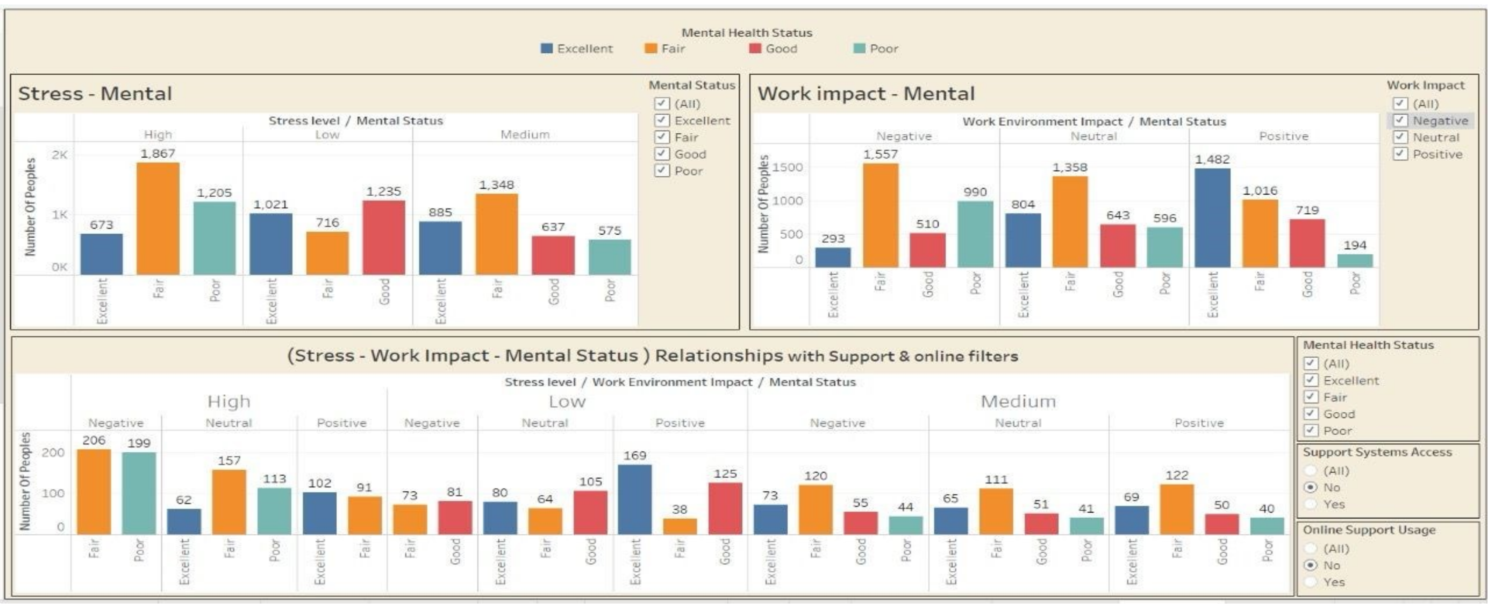
Mental Health Status by Age Groups



This dashboard displays the distribution of mental health status (Excellent, Good, Fair, Poor) across different age groups. It breaks down the data into age intervals such as (14-20, 21-30, 31-40, etc.) to show how mental health varies by different life stages.

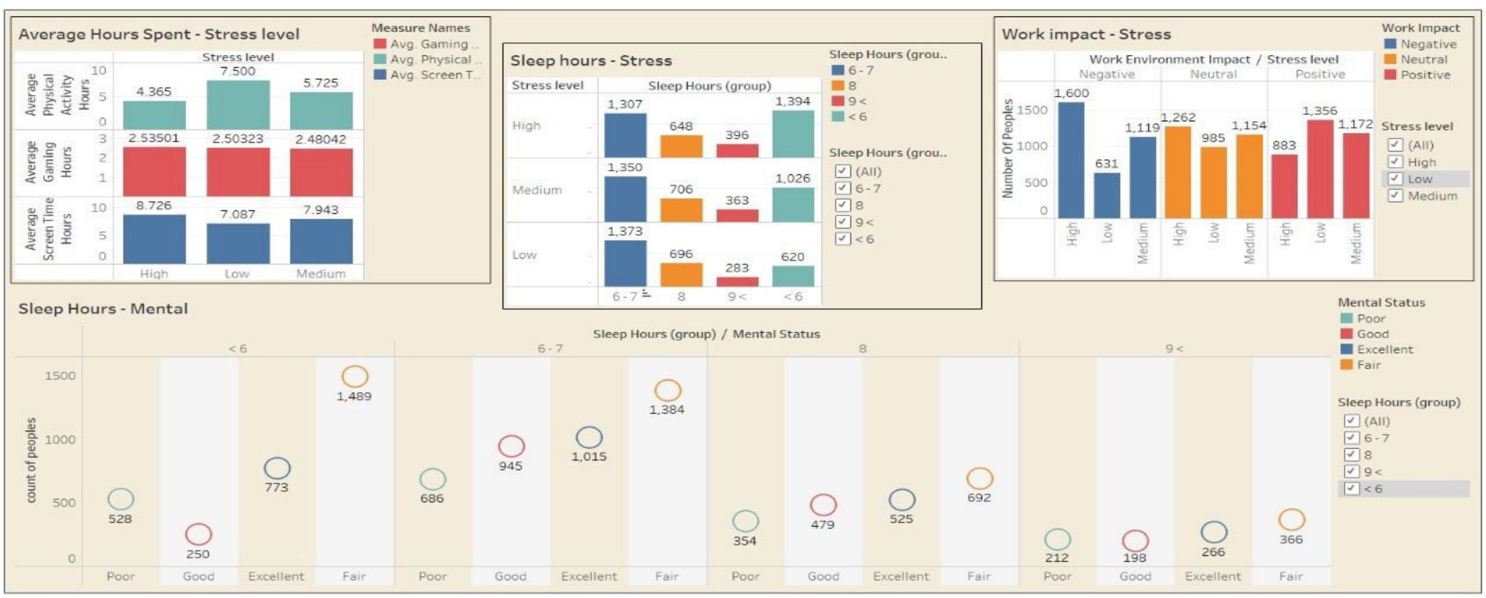


Dashboard1: Impact of Stress and Work Environment on Mental Health



This dashboard illustrates the effect of different stress levels (High, Medium, Low) on mental health status. It also includes a section that shows the impact of the work environment (Negative, Neutral, Positive) on mental health, with further segmentation based on stress levels and the work environment.

Dashboard2: Relationship Between Sleep Hours, Stress, and Mental Health



This dashboard highlights the relationship between sleep hours and stress levels on mental health. It categorizes individuals based on sleep hours (less than 6, 6-7, 8, 9 or more) and links these categories to mental health status. It also includes the influence of time usage on stress and mental health, such as physical activity and screen time.

Screens Time

High-stress individuals spend more time on screens compared to low-stress individuals

Stress level		
High	Low	Medium
8.726	7.087	7.943

Physical Activity

People with lower stress levels engage in more average physical activity hours, while those with higher stress levels are less active

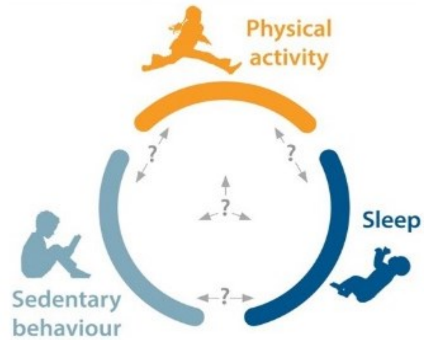
Stress level		
Low	Medium	High
7.500	5.725	4.365

Sleep Hours

People with lower stress levels tend to sleep more (mostly 6-7 hours or more than 8 hours)

Sleep Less Than 6 hours

High	Medium	Low
%46	%34	..



HELPFUL TIPS

Tips for Managers and Employees to Foster a Positive Work Environment

- Create a Positive Workplace Atmosphere -..

- Encourage Open Communication: Make it easy for employees to share their thoughts, concerns, and suggestions. Regular feedback helps build trust and t..

- Recognize and Appreciate Efforts: Simple gestures like acknowledging good work can greatly boost morale. Consider implementing a reward system for achievements,..

- Promote Flexibility: Allowing flexibility in working hours or work-from-home options can help reduce stress and improve job satisfaction.

- Organize Team Activities: Arrange team-building events, lunches, or even short breaks together. It helps in bonding and creates a sense of belonging.

Recommendations Based On Analysis

1. Reduce Stress:
 - Set aside time for daily relaxation by practicing meditation or reading.
 ..

2. Work in a Positive Environment:
 - Enhance your workspace by keeping it organized and clutter-free.
 - Engage in positive communication with colleagues and seek support when needed.

3. Physical Activity:
 - Aim to exercise regularly, such as walking for 30 minutes daily.
 - Physical activities can boost mood and help reduc..

4. Limit Screen Time:
 - Reduce screen time, especially before bed, and try replacing it with other beneficial activities.
 - Take breaks from technology and celebrate small ..

5. Get Quality Sleep:
 - Make sure to get 7-8 hours of sleep each night for proper rest and relaxation.
 - Create a bedtime routine that helps you unwind, s..

Makes a Difference, Even a Small One

- Be There for Each Other: Simple acts of support—like checking in on someone, offering a listening ear, or just sharing a kind word—can have a significant positive impact on mental health.

- Share Encouragement: Whether it's friends, family, or colleagues, sharing encouragement and positivity helps create a more supportive environment. Even small compliments or words of motivation can uplift someone's day.

- Provide Emotional and Practical Support: Offering help when someone is facing a challenge, whether it's with work, a personal issue, or something simple like running an errand, can strengthen bonds and reduce stress.

8. Tools and Technologies Used

- **Google Colab & Python** : Utilized for data exploration and cleaning.
  Key Python libraries include:

  - **Pandas** for data manipulation and handling missing values.
  - **Seaborn** for visualizing outliers using histograms and boxplots.

- **SQL (Structured Query Language)** :Used for merging datasets and executing queries to extract key insights.
- **Tableau** : Employed for data visualization, where you created dashboards to present your analysis findings in a user-friendly format.

9. Conclusion

In this project, we analyzed the relationship between technology usage, mental health, and various lifestyle factors. Through careful data cleaning, integration, and SQL queries, we identified key insights related to mental health categories, stress levels, and the impact of work environments. We discovered how factors like sleep hours, screen time, and physical activity influence stress levels and mental health, both directly and indirectly. Additionally, we utilized Tableau to create a dashboard that visualizes these insights, making the data more accessible and actionable. This comprehensive analysis offers a clear understanding of the interconnected factors that impact mental well-being.