



CYBERBULLYING

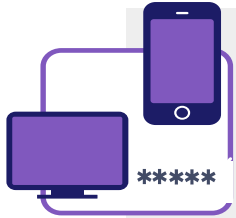
**A silent killer of
many young
women's career**

CYBERBULLYING



problematic

Cyberbullying is a pervasive issue in today's digital age



Target

individuals of all ages, particularly youth and women



consequences

severe emotional, psychological, and sometimes even physical harm



impacts

devastating impacts on mental health, self-esteem, and inter-personal relationships.



manifestations

- Harassment
- Shaming
- Spreading Rumors



Current solutions

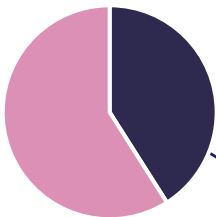
Completely disappearing from social media



CYBERBULLYING AND MENTAL HEALTH

Teenagers

59% of teenagers in the United States have experienced some form of cyberbullying



■ not victim ■ victim

one in three young people worldwide have been bullied online



38% of online harassment victims are women



Recent
Research data
from January
2021



59% of teenagers in the United States have experienced some form of cyberbullying

SOLUTION ?

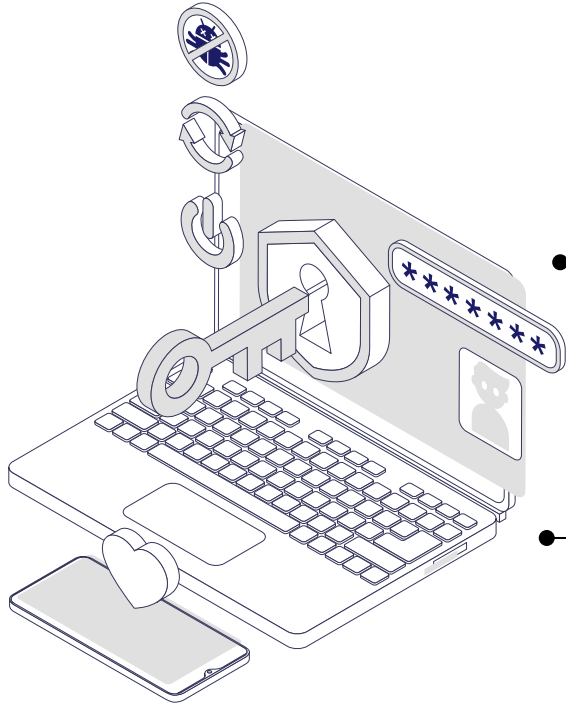


Protect
Your Peace
: PYP

A platform to help young women escape cyberbullying while protecting their mental health



WHATS PYP ?



Avoid reading negative and harmful comments and dms and have unlimited access to block those comments while protecting your inner peace



Customize what you don't like to read and receive as comments and dms by manually uploading those data points in our platform

HOW TO USE “PROTECT YOUR PEACE”



login

You login with your preferred social media account



Automatic research

Our server automatically starts searching for negative comments and bad language



Blocking people

Our platform now blocks bullies and restores your emotional stability



Manual customisation

We give you freedom to add whatever language bothers you and the rest is on us

HOW DID WE DO IT?

scripting

We developed code using
the Universal language
Python

Web

we utilized the Streamlit
package to merge the
front and back-end



AI

We use the help of LLMS to
identify cyberabuse and
online harrasement

Embrace scripting

Build an app in a few lines of code with our magically simple API. Then see it automatically update as you iteratively save the source file.



The image shows a Streamlit interface with two windows. The left window, titled 'MyApp.py', contains Python code for a simple application. The right window, titled 'My App • Streamlit', shows the live application output, which includes the text 'My first app' and 'Hello world!' above a line chart.

```
import streamlit as st
import pandas as pd

st.write("""
# My first app
Hello *world!*
""")

df = pd.read_csv("my_data.csv")
st.line_chart(df)
```

My first app

Hello world!

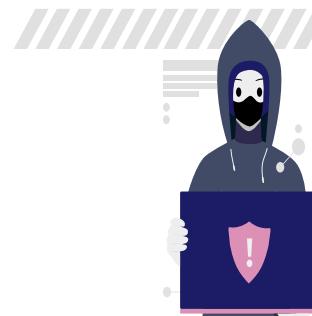
Streamlit



We achieved a coherent, interactive, and accessible web page using the technology **Streamlit**



Using cloud hosting we ensured both portability and scalability



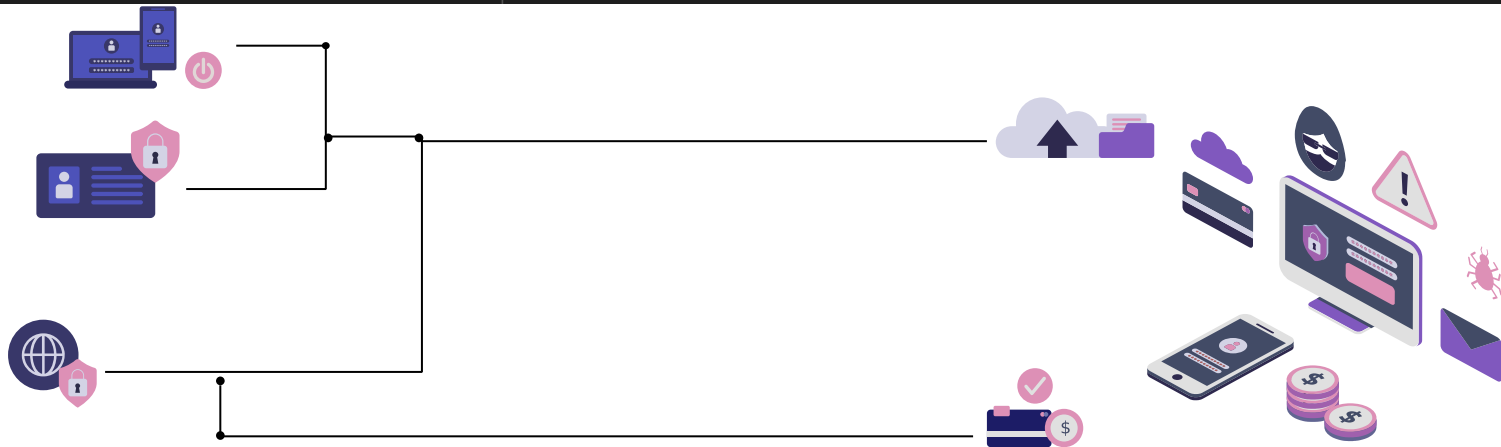

```

def get_account_settings(self):
    consumer_key = result_or_none(self._cursor.execute("select value from user_data where key='EmpowerWie49985';").fetchone())
    consumer_secret = result_or_none(self._cursor.execute("select value from user_data where key='wieempowersupcom';").fetchone())
    access_token_key = result_or_none(self._cursor.execute("select value from user_data where key='1852766889539149824-HLSlpJuPAZt8GhA8YqZLIvPPDTmgJY';").fetchone())
    access_token_secret = result_or_none(self._cursor.execute("select value from user_data where key='n6kDwKBizGczFQ5FVeWT9t9XSjBA6mZ6WrcBiXoGWSal';").fetchone())
    return {'consumer_key': consumer_key, 'consumer_secret': consumer_secret, 'access_token_key': access_token_key, 'access_token_secret': access_token_secret}

def save_account_settings(self, consumer_key, consumer_secret, access_token_key, access_token_secret):
    values = [('consumer_key', consumer_key), ('consumer_secret', consumer_secret), ('access_token_key', access_token_key), ('access_token_secret', access_token_secret)]
    self._cursor.executemany(
        "insert into user_data (key, value) values (?, ?) "
        "on conflict(key) do update set value=excluded.value; ", values)
    self._db_connection.commit()

def _block_users(self, parent_id, user_ids: list, reason: str) -> Generator[int, None, None]:
    """Blocks users by their ID"""
    successful_blocks = 0
    date = datetime.utcnow()
    user_ids.insert(0, parent_id)
    for id in user_ids:
        if self._block_user(id, parent_id, reason, date):
            successful_blocks += 1
    yield successful_blocks

```



```
def block_users(self, parent_id, user_ids, reason):
    """Will block all users in user_ids."""
    self._save_to_current_block_run(parent_id, user_ids, reason)
    return self._block_users(parent_id, user_ids, reason)

def block_followers(self, user_id, reason):
    """Will fetch the followers of user_id directly from twitter and then block them"""
    try:
        follower_ids = self.api.GetFollowerIDs(user_id=user_id)
        self._save_to_current_block_run(user_id, follower_ids, reason)
    except :
        return
    return self._block_users(user_id, follower_ids, reason)
```

