ДОДАТОК А

Програмний код системи

```
DB_FILE = "results.db"
def init_db():
    conn = sqlite3.connect(DB_FILE)
    cur = conn.cursor()
    cur.execute("""
    CREATE TABLE IF NOT EXISTS results (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        timestamp TEXT,
        riasec TEXT,
        abilities TEXT,
        temperament TEXT,
        mbti TEXT,
        recommendations TEXT
    )
    """)
    conn.commit()
    conn.close()
init_db()
def generate_profile_text(riasec, abilities, cluster_id,
cluster_names, top_specs):
    top_riasec = sorted(riasec.items(), key=lambda x: x[1],
reverse=True)[:2]
    riasec_labels = {
        "R": "Практичність, технічне мислення",
        "I": "Аналітичність, дослідницький склад розуму",
        "A": "Креативність, мистецьке бачення",
        "S": "Допомога іншим, емпатія",
        "E": "Лідерство, комунікація, підприємливість",
        "C": "Уважність, точність, організованість"
```

```
    }
    main_traits = [riasec_labels[k] for k, v in top_riasec]
    text = (
        f"На основі ваших відповідей найбільш виражені риси: 
<b>{', '.join(main_traits)}</b>.<br><br>"
        f"Ваша унікальна комбінація здібностей і інтересів 
відкриває найкращі перспективи у сферах, де важливі "
        f"<b>{', '.join(main_traits).lower()}</b>. Це дозволяє 
рекомендувати такі напрямки для подальшого розвитку:<br>"
        f"<b>{', '.join(top_specs)}</b>.<br><br>"
        f"Враховуючи ваш профіль, ви належите до групи 
<b>{cluster_names[cluster_id]}</b>. "
        f"Сучасний ринок потребує саме таких фахівців: 
відкритих до нового, здатних мислити креативно та "
        f"працювати з людьми.<br><br>"
        "Рекомендуємо звернути увагу на розвиток як 
професійних, так і м'яких навичок. "
        "Не бійтеся пробувати себе у нових проектах та брати 
відповідальність за власний розвиток!"
    )
    return text
reference_profiles = [
    # Креативники/мистецтво
    {
        "name": "Стів Джобс",
        "type": "Креативники/мистецтво",
        "vector": [0, 0, 1, 0, 1, 0]
    },
    {
        "name": "Марія Примаченко",
        "type": "Креативники/мистецтво",
        "vector": [0, 0, 1, 0, 0, 0]
    },
    {
        "name": "Біллі Айліш",
```

```
    "type": "Креативники/мистецтво",
    "vector": [0, 0, 1, 0, 1, 0]
},
# Технарі
{
    "name": "Сергій Корольов",
    "type": "Технарі",
    "vector": [1, 1, 0, 0, 1, 0]
},
{
    "name": "Грейс Хоппер",
    "type": "Технарі",
    "vector": [1, 1, 0, 0, 1, 0]
},
{
    "name": "Макс Левчин",
    "type": "Технарі",
    "vector": [1, 1, 0, 0, 1, 0]
},
# Медики
{
    "name": "Катерина Амосова",
    "type": "Медики",
    "vector": [0, 1, 0, 1, 0, 0]
},
{
    "name": "Богомолець Олександр",
    "type": "Медики",
    "vector": [0, 1, 0, 1, 0, 0]
},
{
    "name": "Євген Комаровський",
```

```
    "type": "Медики",
    "vector": [0, 1, 0, 1, 0, 0]
},
# Лідери/менеджери
{
    "name": "Олена Зеленська",
    "type": "Лідери/менеджери",
    "vector": [0, 0, 0, 1, 1, 1]
},
{
    "name": "Сундар Пічаї",
    "type": "Лідери/менеджери",
    "vector": [0, 0, 0, 1, 1, 1]
},
{
    "name": "Сатя Наделла",
    "type": "Лідери/менеджери",
    "vector": [0, 0, 0, 1, 1, 1]
},
# Гуманітарії/соціальники
{
    "name": "Ліна Костенко",
    "type": "Гуманітарії/соціальники",
    "vector": [0, 0, 1, 1, 0, 1]
},
{
    "name": "Василь Сухомлинський",
    "type": "Гуманітарії/соціальники",
    "vector": [0, 0, 1, 1, 0, 1]
},
{
    "name": "Опра Вінфрі",
```

```
        "type": "Гуманітарії/соціальники",
        "vector": [0, 0, 1, 1, 0, 1]
    },
    # Організатори/економісти
    {
        "name": "Крістін Лагард",
        "type": "Організатори/економісти",
        "vector": [0, 1, 0, 1, 1, 1]
    },
    {
        "name": "Джордж Сорос",
        "type": "Організатори/економісти",
        "vector": [0, 1, 0, 1, 1, 1]
    },
    {
        "name": "Владислав Рашкован",
        "type": "Організатори/економісти",
        "vector": [0, 1, 0, 1, 1, 1]
    },
]
def cosine_similarity(a, b):
    return np.dot(a, b) / (np.linalg.norm(a) *
np.linalg.norm(b) + 1e-8)
st.title("Рекомендаційна система освітніх траєкторій ✨")
st.header("Пройди невеликий тест, щоб дізнатись, яка
спеціальність тобі підходить ")
# Блок 1 RIASEC
with st.subheader("Що таке RIASEC? "):
    st.markdown("""
    <div style="background-color: #f0f4ff; padding: 20px;
border-radius: 10px; border-left: 5px solid #2563eb;">
        <h4>Теорія Джона Голланда (RIASEC) класифікує інтереси
особистості на 6 типів:</h4>
```

```
        <ul>
            <li><b>R (Realistic)</b> – практичний, фізичний, технічний (інженер, механік)</li>
            <li><b>I (Investigative)</b> – аналітичний, дослідницький (науковець, IT)</li>
            <li><b>A (Artistic)</b> – креативний, естетичний (дизайнер, митець)</li>
            <li><b>S (Social)</b> – соціальний, допоміжний (психолог, викладач)</li>
            <li><b>E (Enterprising)</b> – підприємницький, лідерський (менеджер, політик)</li>
            <li><b>C (Conventional)</b> – організований, точний (бухгалтер, аналітик)</li>
        </ul>
    </div>
    """, unsafe_allow_html=True)
st.subheader("Обери 3 типи, які тобі найближчі:")
riasec_types = {
    "R": "Практичний / Realistic",
    "I": "Аналітичний / Investigative",
    "A": "Креативний / Artistic",
    "S": "Соціальний / Social",
    "E": "Лідерський / Enterprising",
    "C": "Організований / Conventional"
}
col1, col2 = st.columns(2)
selected_riasec = []
with col1:
    for code in ["R", "I", "A"]:
        if st.checkbox(f"{code}: {riasec_types[code]}", key=f"chk_{code}"):
            selected_riasec.append(code)
with col2:
    for code in ["S", "E", "C"]:
```

```
        if st.checkbox(f"{code}: {riasec_types[code]}",
key=f"chk_{code}"):
            selected_riasec.append(code)
if len(selected_riasec) > 3:
    st.warning("Оберіть не більше трьох типів.")
elif len(selected_riasec) < 3:
    st.info("Оберіть рівно три типи.")
# Блок 2 персональна оцінка
st.markdown("""
<div style="background-color: #f0f4ff; padding: 20px; border-
radius: 10px; border-left: 5px solid #2563eb;">
    <h4>Модель множинного інтелекту Говарда Ґарднера</h4>
    Теорія передбачає, що кожна людина має унікальний набір
інтелектуальних здібностей:
    <ul>
        <li><b>Лінгвістичний</b> – вміння до мов, слів,
текстів</li>
        <li><b>Логіко-математичний</b> – аналітичне мислення,
логіка, розв’язування задач</li>
        <li><b>Візуально-просторовий</b> – уява, орієнтація в
просторі, візуалізація</li>
        <li><b>Музичний</b> – чутливість до звуків, ритмів,
мелодій</li>
        <li><b>Кінестетичний</b> – координація, тілесна
моторика</li>
        <li><b>Міжособистісний</b> – розуміння інших, емпатія,
командна робота</li>
        <li><b>Внутрішньоособистісний</b> – саморефлексія,
самосвідомість</li>
        <li><b>Натуралістичний</b> – зв’язок із природою,
спостережливість</li>
    </ul>
</div>
""", unsafe_allow_html=True)
st.markdown("---")
st.subheader("Оціни свої унікальні здібності")
```

```
st.markdown("""

<div style="background-color: #f0f4ff; padding: 18px 20px 6px 20px; border-radius: 12px; border-left: 5px solid #2563eb; font-size:17px">

<b>У кожної людини – свій унікальний «профіль сильних сторін». Оціни, наскільки тобі притаманні ці типи інтелекту за шкалою від 0 (зовсім не про мене) до 10 (це повністю про мене).</b>

</div>

""", unsafe_allow_html=True)

st.write("")

abilities = {}

col1, col2 = st.columns(2)

with col1:

    with st.container():

        st.markdown("** Лінгвістичний інтелект**")

        st.caption("Вміння добре висловлювати думки, любиш читати, писати, аргументувати.")

        abilities["Лінгвістичний"] = st.slider("", 0, 10, 5, key="ling")

    with st.container():

        st.markdown("** Логіко-математичний інтелект**")

        st.caption("Легко бачиш закономірності, любиш задачі, аналіз, експерименти.")

        abilities["Логіко-математичний"] = st.slider("", 0, 10, 5, key="logic")

    with st.container():

        st.markdown("** Візуально-просторовий інтелект**")

        st.caption("Гарна уява, бачиш рішення образно, полюбляєш малювати або конструювати.")

        abilities["Візуальний"] = st.slider("", 0, 10, 5, key="visual")

    with st.container():

        st.markdown("** Музичний інтелект**")

        st.caption("Відчуваєш музику, ритм, емоції в мелодіях, можеш співати чи грати на інструменті.")
```

```
        abilities["Музичний"] = st.slider("", 0, 10, 5,
key="music")

with col2:

    with st.container():

        st.markdown("** Кінестетичний інтелект**")

        st.caption("Любиш рух, спорт, маєш добру координацію,
відчуваєш своє тіло.")

        abilities["Кінестетичний"] = st.slider("", 0, 10, 5,
key="body")

    with st.container():

        st.markdown("** Міжособистісний інтелект**")

        st.caption("Вмієш знаходити спільну мову, тонко
відчуваєш емоції інших, працюєш у команді.")

        abilities["Міжособистісний"] = st.slider("", 0, 10, 5,
key="inter")

    with st.container():

        st.markdown("** Внутрішньоособистісний інтелект**")

        st.caption("Знаєш свої сильні й слабкі сторони, часто
аналізуєш себе та свої цілі.")

        abilities["Внутрішньоособистісний"] = st.slider("", 0,
10, 5, key="intra")

    with st.container():

        st.markdown("** Натуралістичний інтелект**")

        st.caption("Цікавишся природою, явищами довкілля,
любиш спостерігати за тваринами чи рослинами.")

        abilities["Натуралістичний"] = st.slider("", 0, 10, 5,
key="nature")

# Блок 3 незалежна оцінка

st.markdown("---")

st.header("Темперамент та особистісні риси")

st.markdown("Дай відповідь на 30 простих запитань про себе.
Вибирай найближчу відповідь:")

questions = [

    ("сангвінік", "Мені легко починати нові знайомства"),

    ("сангвінік", "Я люблю бути в центрі уваги"),

    ("холерик", "Я часто беру на себе ініціативу"),
```

```
    ("холерик", "Мене складно зупинити, коли я чимось
захоплений"),

    ("флегматик", "Мені комфортно, коли все відбувається
повільно та спокійно"),

    ("флегматик", "Я рідко проявляю сильні емоції"),

    ("меланхолік", "Мені важко відновитися після невдач"),

    ("меланхолік", "Я схильний довго обдумувати свої дії"),

    ("соціальний", "Я люблю допомагати іншим"),

    ("соціальний", "Я легко розпізнаю настрій інших людей"),

    ("аналітичний", "Мені подобається шукати логіку в
усьому"),

    ("аналітичний", "Мені цікаво працювати з числами,
графіками, таблицями"),

    ("мовний", "Я люблю писати або розповідати історії"),

    ("мовний", "Я легко висловлюю свої думки словами"),

    ("музичний", "Я легко вловлюю ритм, звуки або музику"),

    ("музичний", "Мені подобається співати або грати на
інструментах"),

    ("кінестетичний", "Я люблю рух, спорт або танці"),

    ("кінестетичний", "Я відчуваю себе краще після фізичної
активності"),

    ("інтроверт", "Мені комфортніше наодинці, ніж у великій
компанії"),

    ("екстраверт", "Я отримую енергію від спілкування"),

    ("стресостійкість", "Я легко адаптуюсь до нових
ситуацій"),

    ("стресостійкість", "Я зберігаю спокій у стресових
ситуаціях"),

    ("саморефлексія", "Я часто аналізую свої думки та
емоції"),

    ("саморефлексія", "Я веду щоденник або записую роздуми"),

    ("амбітність", "Я завжди ставлю перед собою нові цілі"),

    ("амбітність", "Я швидко втомлююсь від рутини"),

    ("наглядовість", "Я помічаю деталі, які інші можуть
пропустити"),
```

```
    ("організованість", "Мені подобається все планувати
наперед"),
    ("гнучкість", "Я легко змінюю плани, якщо потрібно"),
    ("послідовність", "Мені важливо доводити справи до кінця")
]
options = ["Так", "Іноді", "Ні"]
col1, col2 = st.columns(2)
scores = {cat: 0 for cat, _ in questions}
answers = []
for i, (cat, qtext) in enumerate(questions):
    col = col1 if i % 2 == 0 else col2
    with col:
        ans = st.radio(
            f"**{i+1}. {qtext}**",
            options,
            key=f"radio_{i}",
            horizontal=True
        )
        answers.append(ans)
        if ans == "Так":
            scores[cat] += 2
        elif ans == "Іноді":
            scores[cat] += 1
answered_all = all(ans in options for ans in answers)
if not answered_all:
    st.warning("Будь ласка, відповідай на всі запитання ")
# Блок 4 мбті
st.markdown("---")
st.subheader("MBTI: Які твердження про тебе?")
st.markdown("""
<div style="background-color: #f0f4ff; padding: 20px; border-
radius: 10px; border-left: 5px solid #2563eb;">
    <h4>MBTI – Типологія Майєрс-Бріґґс</h4>
```

```
    Це модель, що описує особистість через 4 пари
протилежностей:

    <ul>

        <li><b>E / I</b>: Екстраверт / Інтроверт</li>

        <li><b>S / N</b>: Сенсорик / Інтуїт</li>

        <li><b>T / F</b>: Логік / Емпат</li>

        <li><b>J / P</b>: Раціонал / Імпровізатор</li>

    </ul>

    <b>Познач галочкою ті твердження, які тобі справді
близькі.</b>

</div>

""", unsafe_allow_html=True)

st.write("")

mbti_statements = [

    ("E", "Я заряджаюсь енергією серед людей"),

    ("E", "Мені легко заводити нові знайомства"),

    ("E", "Люблю обговорювати ідеї вголос"),

    ("E", "Комфортно виступаю на публіці"),

    ("I", "Втомлююсь від великої кількості спілкування"),

    ("I", "Часто потребую часу наодинці для відновлення"),

    ("I", "Мені простіше писати, ніж говорити"),

    ("I", "Мені потрібен час, щоб довіритись людям"),

    ("S", "Орієнтуюсь на факти і реальний досвід"),

    ("S", "Звертаю увагу на деталі"),

    ("S", "Віддаю перевагу перевіреним методам"),

    ("S", "Ціную практичність більше за новаторство"),

    ("N", "Мене цікавлять ідеї та теорії"),

    ("N", "Часто думаю про майбутнє"),

    ("N", "Люблю шукати сенс і символіку у звичайних речах"),

    ("N", "Бачу можливості там, де інші не бачать"),

    ("T", "Приймаю рішення логічно і раціонально"),

    ("T", "Краще працюю з чіткими правилами, ніж із
почуттями"),
```

```
    ("T", "Вважаю справедливість важливішою за співчуття"),
    ("T", "Вмію відділяти емоції від роботи"),
    ("F", "Важливо враховувати почуття інших у вирішенні
питань"),
    ("F", "Часто думаю, як не образити когось"),
    ("F", "Шукаю гармонію в стосунках"),
    ("F", "Успіх – це коли всі задоволені"),
    ("J", "Люблю мати чіткий план дій"),
    ("J", "Відчуваю дискомфорт від невизначеності"),
    ("J", "Вчасно завершую розпочате"),
    ("J", "Прагну структурувати свій день"),
    ("P", "Люблю імпровізувати, діяти по ситуації"),
    ("P", "Легко підлаштовуюсь під зміни"),
    ("P", "Мені цікаві різні шляхи до цілі"),
    ("P", "Можу змінити план у будь-який момент"),
]
#random.shuffle(mbti_statements)
col1, col2 = st.columns(2)
mbti_score = {"E": 0, "I": 0, "S": 0, "N": 0, "T": 0, "F": 0,
"J": 0, "P": 0}
for i, (code, statement) in enumerate(mbti_statements):
    col = col1 if i % 2 == 0 else col2
    with col:
        answer = st.radio(
            f"{i+1}. {statement}",
            ["Це про мене", "Не зовсім"],
            key=f"mbti_radio_{i}",
            horizontal=True
        )
        if answer == "Це про мене":
            mbti_score[code] += 1
result = ""
for a, b in [("E", "I"), ("S", "N"), ("T", "F"), ("J", "P")]:
```

```
    result += a if mbti_score[a] >= mbti_score[b] else b

st.markdown(f"**Ваш MBTI-профіль:** <span style='font-size:
1.5em; color: #2563eb'>{result}</span>",
unsafe_allow_html=True)

st.markdown("<br>", unsafe_allow_html=True)

col_center = st.columns([1, 2, 1])[1]

with col_center:

    clicked = st.button(

        "Хочу свій унікальний прогноз!",

        key="main_ai_button",

        use_container_width=True

    )

if clicked:

    st.balloons()

    # Вектор ознак

    feature_vector = []

   # RIASEC-блок (6)

    for trait in ["R", "I", "A", "S", "E", "C"]:

        feature_vector.append(1 if trait in selected_riasec
else 0)

    # abilities (8)

    abilities_order = [

        "Лінгвістичний", "Логіко-математичний", "Візуальний",
"Музичний",

        "Кінестетичний", "Міжособистісний",
"Внутрішньоособистісний", "Натуралістичний"

    ]

    for ab in abilities_order:

        feature_vector.append(abilities.get(ab, 0))

    # MBTI-блок (8): E, I, S, N, T, F, J, P

    mbti_pairs = ["E", "I", "S", "N", "T", "F", "J", "P"]

    for mbti_code in mbti_pairs:

        feature_vector.append(mbti_score.get(mbti_code, 0))

    feature_names = [
```

```
        "riasec_R", "riasec_I", "riasec_A", "riasec_S",
"riasec_E", "riasec_C",
        "linguistic", "logical", "visual", "musical",
"kinesthetic",
        "interpersonal", "intrapersonal", "naturalist",
        "mbti_E", "mbti_I", "mbti_S", "mbti_N", "mbti_T",
"mbti_F", "mbti_J", "mbti_P"
    ]
    df = pd.DataFrame([feature_vector], columns=feature_names)
    model_path = os.path.join("models", "model.pkl")
    classes_path = os.path.join("models",
"target_classes.pkl")
    model = joblib.load(model_path)
    target_classes = joblib.load(classes_path)
    probabilities = model.predict_proba(df)[0]
    ranked = sorted(zip(probabilities, target_classes),
reverse=True)
    st.subheader("Рекомендовані спеціальності:")
    for i, (prob, spec) in enumerate(ranked):
        if prob < 0.055:
            continue
        emoji = "🍀" if i > 2 else "🌟"
        boosted_prob = min(prob * 5, 1.0)
        st.markdown(f"{emoji} **{spec}** – ймовірність:
`{boosted_prob:.2f}`")
kmeans_path = os.path.join("models", "cluster_model.pkl")
kmeans = joblib.load(kmeans_path)
# кластер
cluster_id = kmeans.predict(df)[0]
cluster_title, cluster_desc = cluster_names.get(cluster_id,
("?", "?"))
st.markdown(f"""
<div style="background-color:#e0f2fe;padding:18px 16px 12px
16px;border-radius:10px;margin-bottom:20px;">
```

```
  <span style="font-size:1.2em;"><b>Твій профіль найближчий до групи:</b> <span style="color:#2563eb;"><b>{cluster_title}</b></span></span>
  <br><span style="font-size:1em;">{cluster_desc}</span>
</div>
""", unsafe_allow_html=True)
profile_text = generate_profile_text(
    riasec=dict(zip(["R", "I", "A", "S", "E", "C"], riasec_types)),
    abilities=abilities,
    cluster_id=cluster_id,
    cluster_names=["Креативники/гуманітарії", "Технарі", "Організатори", "Медики", "Лідери/менеджери"],
    top_specs=[spec for prob, spec in ranked[:3]]
)
st.markdown(
    f"""
    <div style="background: #e8f0fe; border-radius: 12px; padding: 16px; font-size: 18px">
        {profile_text}
    </div>
    """, unsafe_allow_html=True
)
riasec_vector = [
    1 if "R" in selected_riasec else 0,
    1 if "I" in selected_riasec else 0,
    1 if "A" in selected_riasec else 0,
    1 if "S" in selected_riasec else 0,
    1 if "E" in selected_riasec else 0,
    1 if "C" in selected_riasec else 0,
]
similarities = [
    (prof["name"], prof["type"], cosine_similarity(riasec_vector, prof["vector"]))
```

```
    for prof in reference_profiles
]
best_match = max(similarities, key=lambda x: x[2])
st.info(
    f"Ваш профіль на {int(best_match[2]*100)}% схожий на профіль {best_match[0]} "
    f"({best_match[1]}).", icon="🕵"
)
```

ДОДАТОК Б

Відомість кваліфікаційної роботи

Позначення	Найменування	Дод. відомості
	Текстові документи	
1.	Пояснювальна записка	92 с.
	Інші документи	
2.	Презентаційні матеріали	__ плакатів

.	.	Прізвище та ініціали.	Підп.	Дата			
Розробив		Рязанцев М.С.			Інтелектуальна система рекомендацій для вибору освітньої траєкторії на основі інтересів та здібностей	Шифр групи	Код напр./спец.
Перевірив		Стьопін О.С.				ІТШІ-21-4	122
Н.контр.		Малєєва І.А.				ХНУРЕ кафедра ШІ	
Затв.		Золотухін О.В.					