ETAP 1: Przygotowanie projektu (1 dzień)

1. **Utwórz repozytorium Git** (np. na GitHubie).
2. **Stwórz projekt Django** (django-admin startproject ewybory).
3. **Stwórz aplikację Django** (python manage.py startapp glosowanie).
4. Dodaj podstawowe modele:
    - Kandydat
    - Wyborca
    - Wybory
    - Glos (bez informacji kto na kogo zagłosował!)

✅ **ETAP 2: Realizacja podstawowej funkcjonalności (3–4 dni)**

Modele:

- Wybory – typ wyborów, data startu i końca, liczba głosów do oddania.
- Kandydat – przypisany do danych wyborów.
- Wyborca – użytkownik z możliwością logowania (możesz użyć AbstractUser).
- Glos – relacja do wyborów i kandydatów, bez relacji do użytkownika.

Widoki:

- Rejestracja i logowanie (opcjonalnie dwuetapowa autoryzacja).
- Lista dostępnych wyborów.
- Głosowanie (sprawdzenie czasu, liczby głosów).
- Wyniki (liczba głosów, frekwencja).

✅ **ETAP 3: Tajność głosowania (1 dzień)**

1. **Zaszyfruj głosy** lub po prostu **nie zapisuj kto głosował** – tylko kto otrzymał głos w danych wyborach.
2. W modelu Glos NIE zapisuj user_id.

✅ **ETAP 4: Logger (1 dzień)**

1. Dodaj własną konfigurację loggera w settings.py:
    - Logi do pliku i konsoli.
    - Filtrowanie po poziomach (INFO, WARNING, ERROR).
2. Użyj logging zamiast print.

✅ **ETAP 5: Frontend / estetyka (2 dni)**

1. Użyj **Tailwind**, **Bootstrap**, lub **Django Crispy Forms**.
2. Szablony HTML z extends, include, responsywny wygląd.
3. Wyróżnij wyniki, kandydatów, czas zakończenia wyborów itp.

✅ **ETAP 6: Dodatkowe funkcje (3+ z listy, 2–3 dni)**

Musisz wybrać co najmniej **3 z poniższych**:

- ✅ **Wykresy** (np. Chart.js, Plotly, Recharts w frontendzie) – pokazanie wyników wyborów.
- ✅ **Captcha** – przy rejestracji/logowaniu (np. django-simple-captcha).
- ✅ **PDF z wynikami** – generowany po zakończeniu wyborów (xhtml2pdf, WeasyPrint).
- 🟡 Wysyłka e-mail (np. z potwierdzeniem rejestracji).
- 🟡 Dwuetapowa autoryzacja (django-otp, django-2fa).
- 🟡 Kalendarz z wydarzeniami (np. fullcalendar.js, django-scheduler).

✅ **ETAP 7: Testy automatyczne (1–2 dni)**

1. Dodaj testy jednostkowe dla:
    - logiki głosowania,
    - rejestracji/logowania,
    - ograniczenia czasu głosowania.
2. Użyj pytest-django lub wbudowanego TestCase.

✅ **ETAP 8: Wdrożenie (1–2 dni)**

1. **Wdrożenie na zewnętrznym serwerze (np. Render, Heroku, Railway, Vercel z backendem)**.
2. Konfiguracja:
    - gunicorn, whitenoise, django-environ (do .env),
    - baza danych PostgreSQL,
    - logi na serwerze.

✅ **ETAP 9: Dokumentacja (1 dzień)**

1. **Komentarze + docstringi** zgodnie z PEP257.
2. Wygeneruj automatyczną dokumentację (Sphinx, mkdocs, docstrings).
3. Plik README.md z opisem:
    - jak uruchomić projekt lokalnie,
    - jak wdrożyć,
    - lista funkcji.