מערכת לניהול תמונות מאירועים.

הגדרת פרויקט ברורה, מענה לצורך קונקרטי של מבצעי הפרויקט, שימושי גם לצרכים אחרים (תמונות ילדים מהגן)

סקירת ספרות – חסר "מתחרה" משמעותי – Google photos. מעבר לזה מעבר על כמה מתחרים קיימים שנותנים מענה

מתן מענה לאתגרים מהטכנולוגיה – תיוג ידני "דורס" תיוג אוטומטי של ai.

חסר מדדים כמותיים – כמה תמונות, כמה משתתפים, וכו'. אין דרישות מוגדרות ליכולות שימושיות בעלי האירועים ומשתתפים.

טכנולוגיות- python, node.js, reactjs, atlas mongodb, cloud and local storage.  
flask, opencv

Usecases – משתתף, eventOwner

פרונט – ממשק, צפייה בתמונות, לראות תמונות, ניהול אירוע

Back for frontend – gateway, authentication management

Backend – face detection, photo grouping

Cloud storage – pictures

Mongodb – userdata, eventdata, photometadata, relationships

Flowdiagram – מורכב מידי, לא מדבר על usecases ומדבר על flow קצה לקצה – קשה לעקוב

דורש הרבה עבודה מצד בעל האירוע – תיוג אנשים, "יצירת" משתתפים, בקרה  
מצד המשתתף – flow קל דורש כניסה למערכת תיוג תמונות ומעבר  
חסר usecase של הרשמה, מה קורה בתהליך ההרשמה.  
משתמש יכול להיות גם משתתף וגם בעל אירוע

ERD – participant לא ברור, Photogroup לא ברור למה נדרש

Mockup – ממשק "אדמין" לא נח, אין יכולת לשים participant ללא הרשמה של המשתתף. רק הowner יכול לראות את כל התמונות

אלגוריתם – שימוש ב-opencv. שימוש במונגו ללא הבנה מספקת של ה"יתרונות"  
אבטחת מידע – JWT, הצפנת נתונים רגישים (סיסמה)  
גאנט – חלוקה מאד "טורית" – יצירה של API,DB בהתחלה ובניית פיצ'רים מעלייהם – דורשים כל הזמן חזרה ל-API

**חוזקות:**

1. פרויקט פרקטי, בא לתת מענה לצורך אישי וקונקרטי, הן של מבצעי הפרויקט והן של צורך בשוק
2. התייחסות מספקת לדרישות סקיוריטי, נגישות, וכו'

**חולשות:**

1. אין מדדים לפרויקט - איך יודעים אם מצליח או לא מצליח, סקר ספרות לא מופיע.

2. אין התייחסות לסיכונים ואיך מבצעים - מה קורה אם הPOC לא עובד?.

3. אין כיסוי של כל ה-usecases בתכנית – קשה להגיד אם הכל נכנס ב