**תוכן עניינים**

ארכיטקטורת המערכת......................................................................................2

עיצוב נתונים ופרוטוקולים.................................................................................3

ממשק משתמש................................................................................................4

2. ארכיטקטורת המערכת

2.1 מבט על

המערכת מחולקת לשלושה מודלים עיקריים.

השרת

הסוכן

האתר

השרת והאתר עובדים בצמידות על אותו מחשב. בעת הרצת השרת, האתר גם הוא עולה בצורה אוטומטית.

הסוכן, הנמצא על מחשבים אחרים (או אותו אחד), שולח הודעות לשרת.

פירקתי את המודלים כך כדי שכל רכיב מערכת יהיה תפקיד עיקרי. השרת אחראי על עיבוד מידע, הסוכן אחראי להעברתו לשרת, והאתר אחראי להציג את התוצאות למשתמש.

2.2 פירוט רכיבי המערכת  
**השרת**

השרת מתחיל את ריצתו על העלתו של האתר כ process נפרד ואז ממשיך בריצתו. (עשיתי זאת כדי שיהיה צורך להריץ רק קובץ אחד – השרת. זה בעיקר לנוחיות. אפשר להריץ את שניהים לכוד והפרוייקט יעבוד אותו הדבר).  
השרת פותח thread שירוץ לאורך שאר ההרצה. תפקידו הוא להאזין לסוכנים חדשים. ברגע שסוכן פונה לשרת, הוא מייצר class שמכיל בשדותיו ערכים המאפיינים את הלקוח ו thread שיטפל בצרכיו וממשיך לחכות לסוכנים נוספים.

thread הסוכן מחכה לקבלת סיכום התעבורה. לאחר שקיבל, מעדכן מספר משתנים לעיבוד עתידי ואז מחכה לסיכום התעבורה הבא. מטרת thread זה הוא פשוט לקלוט מידע ולאגור אותו.

בחזרה ל main – לאחר שפתח את הthread לקבלת סוכנים חדשים, נכנס ללולאה בה בוחר מתי לייצר דוח חדש (על פי הגדרות שקיבל בתחית הריצה כארגומנטים).

כאשר הגיע הזמן ליצור דוח, מדפיס את מספר החבילות הכולל שסוכמו עד כה ומתחיל לייצר את הדוח עצמו ושמירת נתונים חדשים ל DB.

**הסוכן**

הסוכן הינו מאוד נאיבי. מטרתו היחידה היא להסניף תעבורה ולסכם אותה.

הסוכן מתחיל לאגור פקטות (בשימוש של scapy) עד שמגיעה למספר ההוגדר לו (מועבר כארגומנט). כשהגיע לכמות הדרושה, מתחיל לעבור עליהם אחד אחד, ומייצר class בעל שדות מוגדרת מראש המכיל פרטים על הפקטה. לשם מילויי ה class יש פונקציה יחודית שמטרתה למלאות אותו. הסוכן מכין רשימה הבנוייה מהסיכומונים וכשהגיע לסוף, שולח את הרשימה לשרת ומתחיל את התהליך מהתחלה.

**האתר**

האתר הוא החלק היצוגי של הפרוייקט. הוא רשום בשימוש של flask. כאשר משתמש מתחבר לאתר, מוצג בפניו עמוד הבית. שם יכול לבחור להסתכל על דוחות שונים שהשרת ייצר. בנוסף, האתר מייצר סיכום דוחות בשימוש של ה DB שהשרת אגר.

2.3 דיון בנושא העיצוב הנבחר

היתרון בחלוקת הרכיבים לצורה הזו היא שמאוד קל להרחיב ולשנות כל רכיב. ניתן להוסיף שדות נוספים לכל פקטה בקלות, להוסיף עוד עמודים לאתר וניראות הדוחות שהשרת מציג. ניתן גם לשנות את הסוכן השרת והאתר לכל שפה רצויה והפרויקט ימשיך לעבוד כראוי כל עוד צורת המידע (פרוטוקול התקשורת) לא משתנה.

החסרון הוא שלפעמים יש קוד נוסף המכין הרחבה עתידית שלעולם לא ישומש. כלומר כתיבת קוד לא דרוש.

1. עיצוב נתונים ופרוטוקולים

**פרוטוקול תקשורת**

בפרוייקט יש 2 פרוטוקולים ברשת.  
בין הסוכן והשרת:

הסוכן שולח לשרת רשימה בכל איבר הוא מילון המכיל 6 שדות, כל מילון מייצג סיכום של חבילה. המילון מכיל- שם תוכנה (מחרוזת), כתובת יעד (מחרוזת), מיקום גלובלי/מדינה (מחרוזת), האם החבילה נכנה או יצאה מהמכשיר (בוליאני), פורט בצד הסוכן (מספר) וגודל החבילה (מספר).

בין המשתמש לאתר:

שימוש סטנדרתי של פרוטוקל http- האתר שולח למשתמש את עמוד ה html.

**בסיס נתונים**

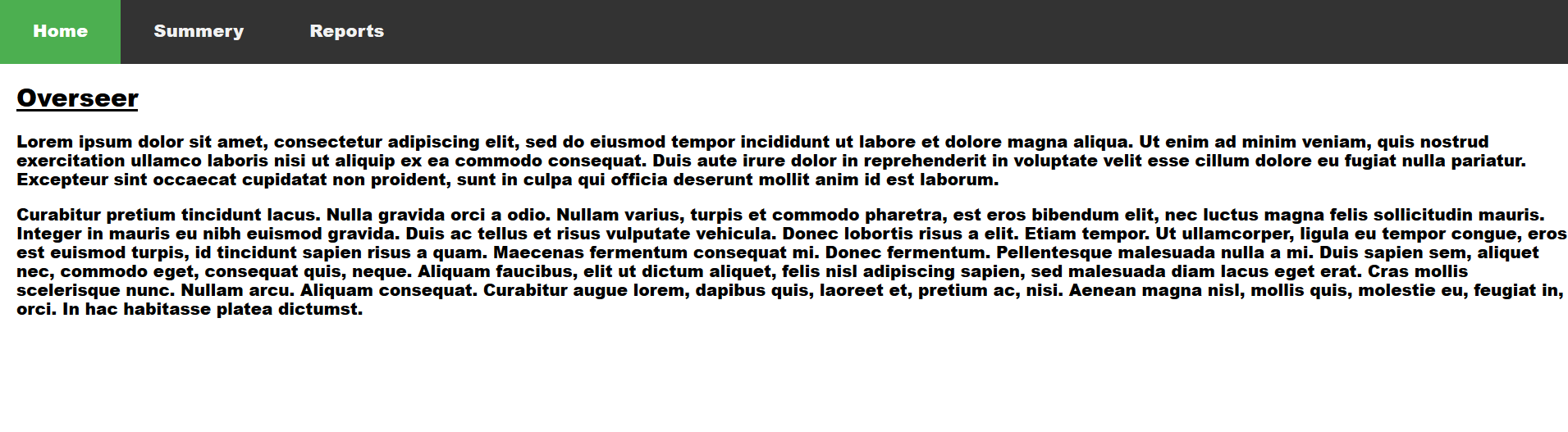
בסיס הנתונים בו אני משתמש הינו קובץ סטנדרתי בעל סיומת .dat – הקובץ שומר מחרוזת json של מילון, המכיל כמפתח את שם הדוח וכערך רשימה המכילה – מילון שמכיל כמפתח כתובת מכשיר וכערך את מספר בתים שקיבל, מילון שמכיל כמפתח כתובת מכשיר וכערך את מספר בתים ששלח, מילון שמכיל בכמפתח כתובת מכשיר וכערך התראות (כמו חיבור או התנתקות), מילון המכיל כמפתח מיקום גולובלי/שם מדינה וכערך את גודל המידע שנשלח לשם, מילון המכיל כערך את כתובת היעד וכערך גודל מידע, מילון המכיל כמפתח שם תוכנה וכערך גודל מידע, מילון המכיל כערך מספר פורט וכערך גודל מידע.

4. ממשק משתמש

ממשק המשתמש בפרויקט שלי הוא למעשה האתר. לאתר יש 4 מסכים עיקריים.

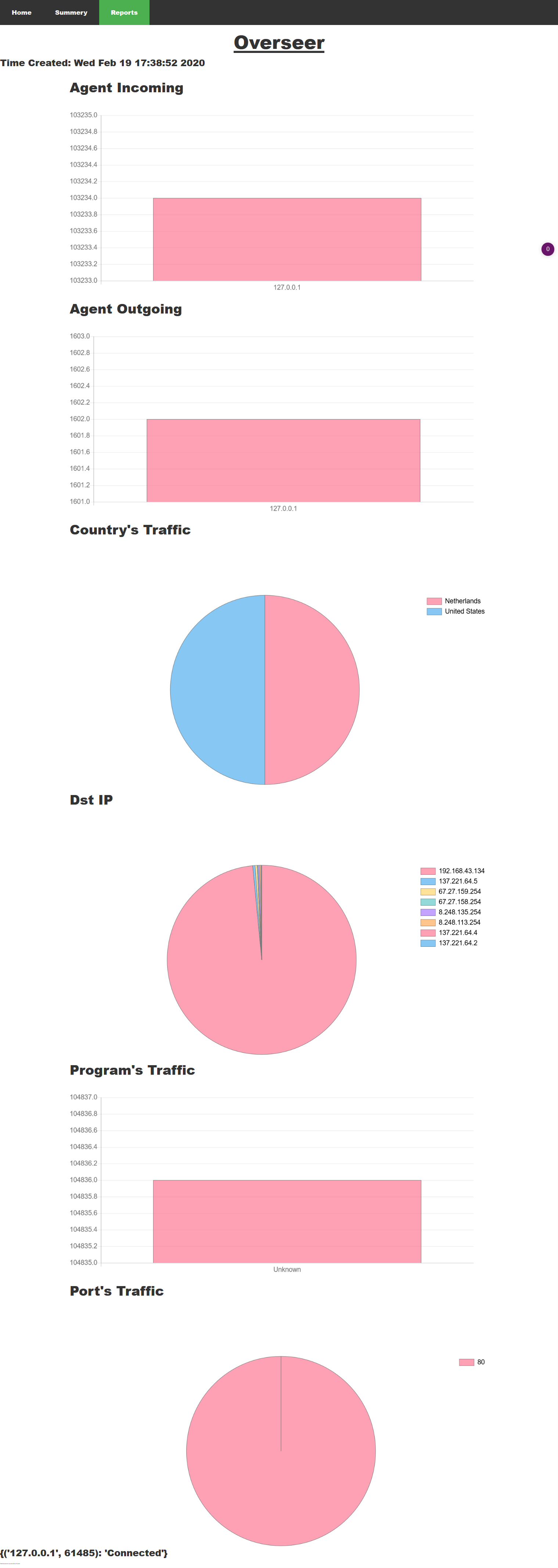
בכל מסך יש מעין tabs בחלקו העליון. הוא נועד למעבר בין חלונות נחויות.

בית



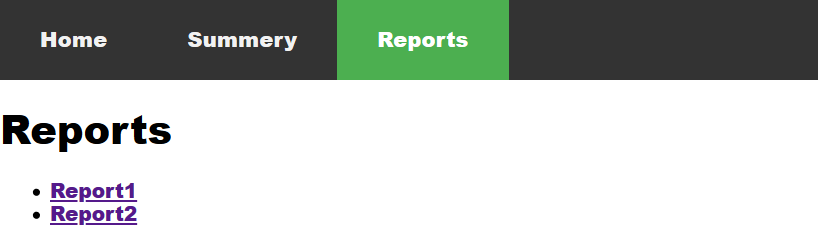
כך נראה מסך הבית. (כרגע יש פה גיבריש שמשמש כ filler. זה יוחלף בתיאור הפרויקט)

דוח

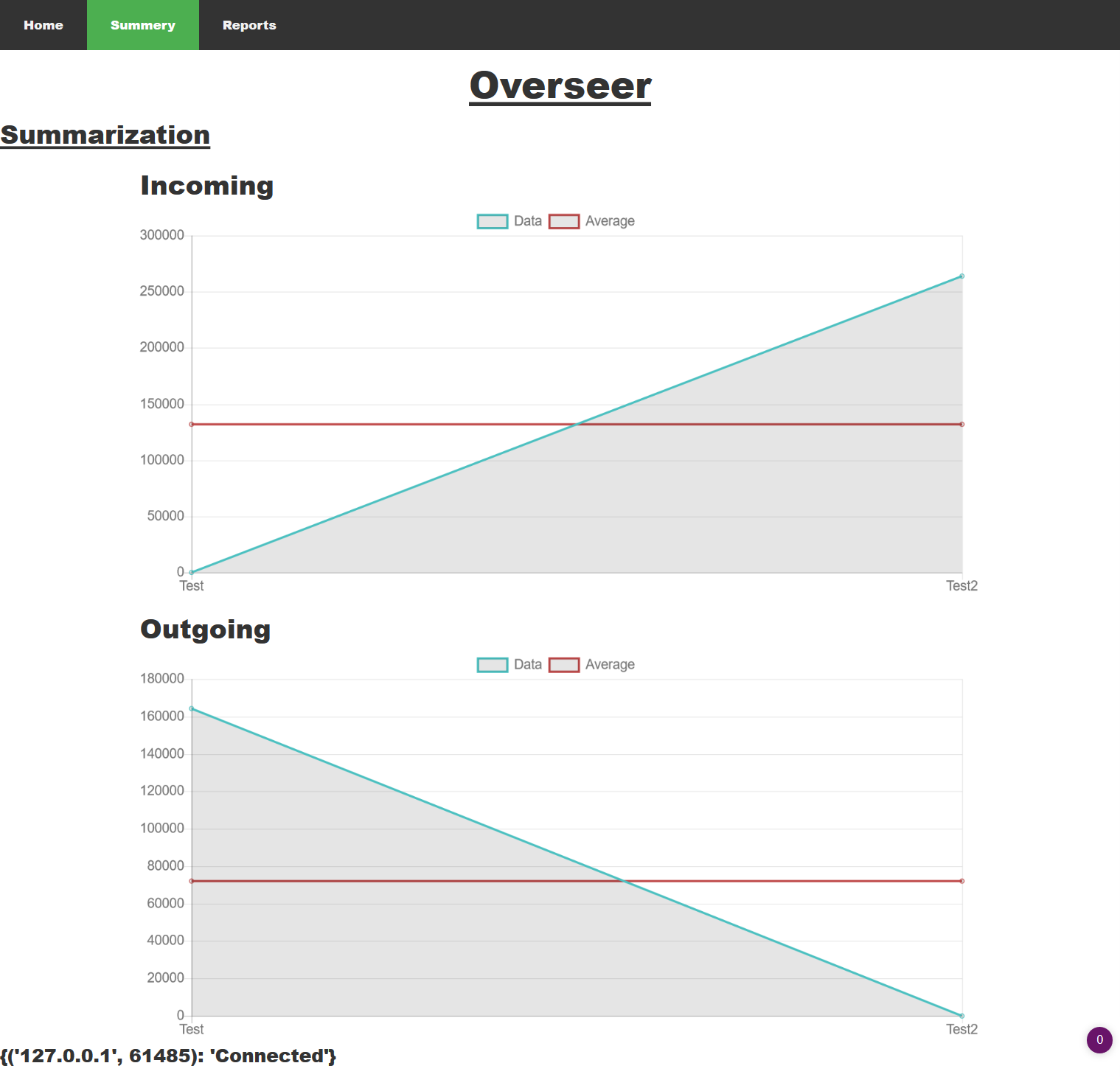


כך נראה הדוח שהפריקט מייצר.

רשימת כל הדוחות

במסך זה יופיעו כל הדוחות שיוצרו.

סיכום

מסך זה מציג סיכום בגרפים לכל הדוחות שיוצרו בעזרת ה DB.