Akela Overlap

ברוך הבא לצוות אקילה!

צוות אקילה מפתח ומקדם את פרויקט אקילה המורכב משתי מערכות:

1. AkelaDex – אפליקציית Web מבוססת React.
2. AkelaApp – אפליקציית Android מבוססת React Native.

המערכת משמשת את הקצה המבצעי בעת פעילויות מבצעיות שונות, בדגש על מעצרים, באיו"ש.

ועם קצת פחות סיסמאות ורשמיות – המערכת בשימוש ע"י לוחמים בעת מעצרים של מבוקשים בשטחי הרשות הפלסטינית, ובכך תורמת באופן ישיר ומובהק לביטחון המדינה.

אנחנו בצוות בטוחים שאינך יכול לחכות לתרום לביטחון המדינה באופן ישיר,

אך לפני זה, עלינו להיות בטוחים שאתה יודע מה לעשות, ולכן, עליך לבצע חפיפה מקצועית לפני שתפיל לנו את הProduction.

החפיפה הורכבה כך שתכיל את כל הנושאים מהבסיסיים ביותר, לכן חשוב לנו בצוות לשמוע את הפידבק שלך.

אל תתבייש לבקש מהחונך לדפדף נושאים מסוימים שאתה מרגיש שולט בהם, זאת בהינתן שהחונך באמת יוודא שאתה שולט בהם, כמובן.

צוות אקילה מאחל לך המון הצלחה וסומך עלייך שתסיים את החפיפה זריז!

1. **הישגים נדרשים:**

- שליטה מקצועית בטכנולוגיות ושפות הפיתוח של הפרויקט.
- עבודה על פרויקט בעזרת Git ושימוש בכלל הכלים הרלוונטיים אשר הוא מציע.
- התיישרות לסטנדרט כתיבת הקוד **המחמיר בטירוף** של הצוות ושמירתו בקנאות.
- התנסות במתודולוגיית Agile על כל הכלים והטקסים שהיא מכילה.
- הכרת ענף "אבן מתגלגלת" על כלל מוצריו ומדוריו לעומק.
- הגברת תחושת המסוגלות לפיתוח תכולות מבצעיות.

2. **הליך החפיפה:**

- תיאום ציפיות לחפיפה עם החונך.
- העמקה מקצועית בשפות הפיתוח Js, Ts.
- עקרונות SOLID וClean Code.
- תיאום ציפיות לפרויקט ולמידת עקרונות Agile.
- למידת טכנולוגיות צד שרת וDB.
- כתיבת צד השרת לפרויקט החפיפה.
- למידת טכנולוגיות צד לקוח.
- כתיבת צד הלקוח לפרויקט החפיפה.
- סיכום החפיפה.

3. תיאום ציפיות לחפיפה:

- תשובה לכל שאלה נמצאת אצל הרב גוגל, אם מושג או תחום לא מובן מספיק מצופה ממך לשרוף את המרשתת לפני שתשאל את הצוות.
- בניגודיות מוחלטת לנקודה שמעל, תרגיש חופשי לפנות לצוות בנוגע לכל שאלה, אנחנו כאן כדי לחנוך וללמד אותך.
- עד כמה החפיפה תהייה עמוסה ומעמיקה תלוי לגמרי בך, אם תרגיש שאתה בטוח בנושא מסוים תרגיש חופשי לרפרף עליו, כמובן בשיח עם החונך.
- צוות אקילה מצפה מהתוכניתנים שלו להשפיע על המוצר.

תגדיל ראש לגבי התכולות של הפרויקט חפיפה ותשקיע בעיצוב.

- לעיתים זה ירגיש שלכאורה אין עלייך לחץ. נהפוכו. הצוות מחכה שתיתן כתף בכמות תכולות לא נגמרת שהמוצר מפיל עלינו, כמה שתיתן מעצמך בתקופה הזאת יהפוך אותה לקצרה יותר ויכניס אותך לצוות בצורה מהירה הרבה יותר, מקווים שתשבור את השיא!

4. העמקה מקצועית בשפות הפיתוח:

Javascript

ביצוע קורס Udemy שיסופק ע"י הצוות.

יש לבצע את פרקים 1 – 10. **לא כולל פרקים 5, 7, 8.**

יש לתת דגש וקשב לנושאים הבאים:

- Lexical scope, Lexical this
- Async/Await and Promise
- ES6 Features

Typescript

העמקה מקצועית בדוקומנטציה של Ts, ומענה על השאלות הבאות:

1) למה צריך להשתמש בTs?
2) האם אנו מחויבים להשתמש בכלל החוקים שהשפה דורשת?
3) כיצד ניתן להימנע מחוק מסוים? ספק כמה דוגמאות.
4) סכם על סוגי הTypeים הקיימים.
5) ספק הסבר קצר על המושגים הבאים:
    - Union
    - Generic
    - Intersection
    - Keyof
    - Typeof
    - Partial
    - Required
    - Readonly
    - Record
    - Pick
    - Omit

5. **עקרונות SOLID וClean Code:**

אנחנו בצוות אובסיביים לסטנדרט קוד גבוה, אבל לפני שתתחיל לקבל כאפות בMerge Requests, עלייך להבין **למה** זה כל כך חשוב לנו.

אז תכין פופקורן, תפנה שעתיים בלו"ז ותחפש ביוטיוב את הסרטון הבא:

Clean Code: The Next Chapter by Victor Rentea

אחרי הצפייה המהנה, עלייך לסכם בקצרה על כל אחד מעקרונות הSOLID.

6. **תיאום ציפיות לפרויקט ולמידת עקרונות Agile:**

- את הפרויקט תבצע בהתאם לכל עקרונות פיתוח Agile לאחר שהוסברו לך על ידי החונך.
- הסבר כללי לפרויקט קיים במסמך, אך המשימות יאופיינו בפלטפורמת ניהול המשימות הצוותית – TFS.
- כנאמר לעיל במסמך, האפיון לפרויקט הוא בסיסי ושטחי. האיכות של הפרויקט תלויה בך מקצה לקצה. תשקיע בחשיבה על התכולות ודרכים לשפר אותן. תשקיע בעיצוב ובראיית מוצר.

תוכל לקבל השראה מאפליקציות פופולריות קיימות או מחיפוש זריז בגוגל לכל תכולה שהיא.

כל זה מצופה ממך ונדרש ממך לא פחות מכתיבת קוד איכותי.

- עבודה על כל תכולה תתבצע בהתאמה מדויקת לפי עבודתו של תוכניתן בצוות בדגש על כתיבת טסטים, כתיבת DoD, תכנון ותמחור המשימה וניהולה בBoard.
- כתיבה שוטפת לRetro החפיפה על מנת שבנחפף הבא נוכל ללמוד ולהשתפר!

7. פרויקט חפיפה – AkelaEven

הצורך המבצעי:

מרוב כל התקנים שענף "אבן מתגלגלת" **הרוויח ביושר** בשנתיים האחרונות נוצר מצב של חוסר שליטה באנשים המאיישים את התפקידים.

על כן, הדרג הניהולי הבכיר החליט לחפש תוכניתן גיבור אשר יוכל לתת מענה למיפוי כלל המשרתים בענף עם כמה שיותר מידע על כל אחד מהם.

המענה המבצעי:

עליך לייצר מערכת עם אפליקציות Web וMobile אשר בעזרתה נוכל למפות את הענף על כלל בעלי תפקידיו ומיקומם.

המערכת תתוכנן באופן הבא:

- אפליקציית Mobile מבוססת React Native– אשר תכיל אפשרות להתחברות, הרשמה, תיעוד וצפייה בתיעוד שיכילו:
  1) צילום של המשרת
  2) מידע בסיסי – שם, דרגה, תפקיד, כתובת
  3) תיעוד של משרד המשרת.
  4) **בונוס מארץ הבונוסים** – השאלה שלו בשאלון.
- אפליקציית Web מבוססת React – אשר תכיל אפשרות להתחברות, הרשמה, צפייה בכל התיעוד שצולם במובייל בשילוב BI על הנתונים.
- צד שרת מבוסס NestJs – אשר יכיל את כל הדגשים שנלמד בנושא לרבות התממשקות לשירות MinIO לאחסון התמונות ושמירת כלל מידע בMongoDB תוך שימוש באגרגציות לשליפתו.

8. למידת טכנולוגיות צד שרת וDB:

NestJs

קריאת הדוקומנטציה של הספרייה החלומית הזאת ומענה על השאלות הבאות:

- מה זה ExpressJs? מה הקשר בינו לבין NestJs?

Controller

- איך מגדירים Route?
- לאילו אלמנטים של הבקשה ניתן לגשת מהפונקציות בController? איך אפשר לבצע זאת?
- איך מחזירים סטטוס מסוים עבור Route?
- כמה מופעים נוצרים לController באפליקציה? האם בקשות שונות חולקות את אותו המופע?

Provider

- אילו סוגים של אובייקטים יכולים להוות Provider?
- מהו Provider Token?
- תן דוגמה לשימוש בuseFactory.

Modules

- הסבר בקצרה את המושג Module. למה הוא משמש?
- כמה Modules יש באפליקציה של Nest?
- ציין את הproperties הקיימים באובייקט ש()Module@ מקבל כפרמטר, מה שמעותם?
- הסבר על המושג ()Global@

Tests

- קרא באינטרנט על סוגי הטסטים השונים, הסבר בקצרה על Unit Test, E2E Test, Intergration Test.
- למה שנשקיע את הזמן שלנו בטסטים? ציין שלוש סיבות לפחות.
- קרא בהרחבה על טסטים בNest, על איזו ספריית טסטים הכי ממליצים?

MongoDB

עלייך לבצע קורס מונגו אשר יסופק לך ע"י החונך.

9. פרויקט חפיפה – צד שרת

אחרי שלמדת על טכנולוגיות צד השרת שלנו בצורה בסיסית, עליך לייצר את צד השרת של פרויקט החפיפה.

שם לב שמצופה מהמערכת שלך לדעת לקבל בקשות בהתאם לנדרש ממך בפרויקט ולנהל אותן.

ההגדרה של הסיום והמשך לשלב הבא בחפיפה היא האפשרות לפנות לשרת ולקבל ממנו תשובה על כלל הבקשות הנדרשות.

דגשים לכתיבת צד השרת:

- שימוש מעמיק בכלי Typescript, כולל תכנון נכון של כלל האובייקטים שידרשו גם לצד לקוח.
- שימוש בValidation Pipe בכל Route של המערכת.
- מימוש Guard שאוכף JWT בכל Route של המערכת.
- מימוש Middleware של לוגים, תיעוד של כל פעולה שמתבצעת וסיווג הLog לפי קטגוריה: שגיאה, אזהרה, מידע.

- שמירה בקנאות על השכבות ומימוש כל לוגיקה במקום ההגיוני שלה.
- התממשקות לשירות MinIO ושמירת התמונות בעזרתו.
- E2E Test לכל Route.
- Unit Test לכל פונקציה בService.
- יצירת סביבות פיתוח ומשתני סביבה בקבצים רלוונטיים.
- תפיסה, טיפול ותיעוד של כל שגיאה.
- כל הקוד מנוהל בGit.
- סטנדרט קוד של השמחות לרבות:

1) שמירת Constים גלובאליים במקום רלוונטי.
2) קובץ Utils עם פונקציות עזר גלובאליות.
3) אין בשום מקום במערכת @ts-ignore, ולא as (אופרטור של ts) בלי הסבר מטורף ללמה היית צריך.
4) פונקציות קצרות שמבצעות פעולה אחת בלבד בלי שום אפקט צדדי.
5) שמות משתנים רלוונטיים.

אישורי תוכניות: (אחריותך לקבוע עם החונך)

- אישור תכנון אובייקטים מול החונך.
- אישור תכנון מודולים מול החונך.
- אישור תכנון עץ תיקיות מול החונך.

10. למידת טכנולוגיות צד שרת וDB:

React

העמקה בדוקומנטציה של ריאקט ומענה על השאלות הבאות:

1) מה הייחודיות והיתרון של ספריית ריאקט לפיתוח Web?
2) הסבר בקצרה לתפיסתך את הקונספט של קומפוננטה.
3) הסבר בקצרה על Hooks, הסבר בהרחבה על useState וuseEffect.
4) תן דוגמה למתי אשתמש בuseCallback והסבר מה יקרה אם לא אשתמש.
5) הסבר בהרחבה על HOC, תן דוגמה לHOC.
6) הסבר על Redux בהרחבה, איזה מענה אנו מקבלים דרכו לאיזו בעיה.
7) הסבר על Saga, מה מטרתו ואיך הוא משתלב בStore שלנו.
8) מה ההבדל בין Redux State לuseContext? ציין דוגמאות לשימוש בכל אחת מהן.
9) מה ההבדל בין props לstate בקומפוננטה?
10) הסבר בקצרה על המושג React Fragments.

React Native

תפנה לעצמך יומיים בכיף כדי להרים סביבת פיתוח של נייטיב במחשב.

תתכונן לסבל.

לאחר הקמת סביבת הפיתוח, הכן מצגת לימודית להסברה לצוות על כלל התהליכים שמתבצעים המאפשרים כתיבת קוד Javascript והרצתו על מנוע Android וiOS במקביל.

11. פרויקט חפיפה – צד לקוח

אתה מתקרב בצעדי ענק לסיום החפיפה, יפה מאוד.

עכשיו הגענו לחלק האמנותי, פיתוח צד הלקוח של הפרויקט שלך.

לשם כך תצטרך לפתח אפליקציות Web וMobile אשר כל אחת תבצע את תפקידה לפי האפיון שיהיה קיים בTFS ויסופק לך ע"י החונך שלך.

בסוף זמן הפיתוח המערכת צריכה להיות פונקציונלית ועובד, ועלייך להשתמש בה על מנת לתעד את כל הענף, אז לטובתך עדיף שלפחות החווית משתמש שלה תהייה ברמה גבוהה.

דגשים לכתיבת צד הלקוח:

- צור את הפרויקט בעזרת create-react-app עם טמפלייט Typescript בלבד.
- כתיבת קומפננטות פונקציונליות בלבד עם הסטנדרט שקיים כרגע בצוות.
- שימוש בhooks בחוכמה ע"פ מה שהקומפוננטה דורשת, יהיה עלייך לדעת להסביר גם למה השתמשת.

- שימוש בRedux Persist במובייל על מנת לשמור את הState גם כשאפליקציה נסגרת.
- שימוש בreact-native-offline על מנת שהאפליקציה תוכל לעבוד בנתק.
- שימוש בreact-hook-form לכל הinputים כולל ולדיציות של הקלט.
- עיצוב קומפוננטה התבצע אך ורק עם אובייקט Style ובשום אופן לא עם css.
- פיצול אגרסיבי ונכון של קוד בקומפוננטות. לא להתבייש.
- בלי רנדורים מיותרים, נבדוק את זה.
- שימוש בRedux, חנויות שבנויות בתבנית מוגדרת של הסטנדרט שקיים כרגע בצוות.
- בכל עבודה עם תאריכים וזמנים אך ורק moment.js.
- להשתמש בreact-navigation במוביל ובreact-router בweb.
- אין שום warning/error בconsole אלא אם כן יש הסבר סוחט דמעות.
- כמה שפחות לוגיקה בעצמך, לא לפחד להשתמש בספריות קיימות, יחד עם זאת להסביר למה דווקא הספרייה הנבחרת ולא המתחרים.
- קבצי Config וUtils בידיוק כמו בשרת, כולל HttpConfig שמנהל את כל הבקשות.
- קבועי מערכת בקבצים מסודרים, כולל צבעי המערכת שנקבעו.
- סביבות פיתוח ומשתני סביבה.

- חיווים רלוונטיים למשתמש על כל שגיאה לרבות קלט לא נכון ופופאפים של שגיאות.
- שימוש בספריית Material-UI בweb וב React Native Paper במובייל בשביל UI נפץ.
- השקעה נרחבת בעיצוב וחווית משתמש.

אישורי תוכניות: (אחריותך לקבוע עם החונך)

- אישור תכנון קומפוננטות מול החונך.
- אישור תכנון עיצוב מול החונך.
- אישור תכנון עץ תיקיות מול החונך.