ברוך הבא לצוות אלפא

צוות אלפא מפתח את אפליקציית אלפא – מערכת שליטה ובקרה שעוזרת להתנהלותם של כוחותינו בשטח בכל רחבי הארץ.

<u>אבל מה יש בתכלס?</u>

יש לנו מערכת מובייל שמציגה מפה ומראה תמונת מצב בזמן אמת – ניתן לראות את כוחותינו בשטח, דיווחים בזמן אמת, מידע סטטי ממקורות מידע שונים ועוד.
המערכת בשימוש ע"י משתמשים שונים, חלקם לוחמים בשטח וחלקם נמצאים בפיקודים.
אנחנו נותנים פתרון נוח וזמין לבעיה של אי ודאות לגבי מצב השטח ברגע נתון:
המערכת הזו מצילה חיים!

אז בגלל שזו מערכת קריטית מאוד כתבנו לך חפיפה פסיכית! היא תהיה מורכבת מחלקים עיוניים הכוללים קריאה של דקומנטציות שונות ופרוייקט חפיפה שיסגור אותך טוב טוב על הדברים.

אל תתבייש להיעזר בחונך שלך ובשאר הצוות אם יש דברים שלא ברורים לך עד הסוף – כולם ישמחו להשוויץ קצת בידע שלהם...

אז יאללה סומכים עליך שהחפיפה תסתיים מהר :)
לא יכולים לחכות!

תהליך החפיפה

<u>הישגים נדרשים:</u>

- 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

<u>הליך החפיפה:</u>

- למידת טכנולוגיות צד שרת וDB
- כתיבת צד שרת לפרויקט
- למידת טכנולוגיות מובייל
- כתיבת מערכת מובייל לפרויקט

כל שלב ושלב ילווה בסקרי קוד (בגיט) שהצוות יעביר

<u>תיאום ציפיות לחפיפה:</u>

- מצופה ממך לשרוף את המרשתת לפני שתשאל את הצוות ,תשובה לכל שאלה נמצאת אצל הרב גוגל, אם מושג או תחום לא מובן מספיק שאלה, אנחנו כאן כדי לחנוך וללמד אותך
- בניגודיות מוחלטת לנקודה שמעל, תרגיש חופשי לפנות לצוות בנוגע לכל בטוח בנושא מסוים תרגיש חופשי לרפרף עליו, כמובן בשיח עם החונך
- עד כמה החפיפה תהייה עמוסה ומעמיקה תלוי לגמרי בך, אם תרגיש שאתה
- תגדיל ראש לגבי התכולות של הפרויקט חפיפה ותשקיע בעיצוב קוד
- 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.

איטס שואו טיים!

אז כחלק מהחפיפה נרים מערכת שאמורה לכסות את כל העקרונות שאנחנו ממשים במערכת – נעשה את זה ונהיה מסודרים.

<u>פרוייקט חפיפה – נכון במכון</u>

הצורך המבצעי

כל חייל וחייל במדור 43 מתאמן בזמנו הפנוי (אין מה לעשות זה המצב) ולכן החלטנו להרים מערכת שתעזור לחיילים להיות בבקרה על האימונים שלהם ולמצות אותם כמה שאפשר.

איך נעשה את זה?

ניצור אפליקציית מובייל אשר בעזרתה נוכל לתעד ולתכנן את האימונים שלנו לעומקם.

המערכת תיתן את היכולות הבאות:

- לצפות בתיעוד של אימונים שעשיתי בעבר
- לתעד אימון לפרטים (נתעד כל תרגיל, סט, משקל, זמן במאמץ וכל דבר אחר שנוכל לחשוב עליו)
- לתכנן אימונים (לוח שנה ובו נוכל לתכנן מראש)

כמובן שכל המידע מהטלפון נשמר בשרת ונוכל לקבל אותו גם ממכשירים אחרים.

לפני שמתחילים...

נרצה שתעבור על כמה נושאים כלליים ולוודא שאתה מבין אותם היטב

Javascript

ביצוע קורס javascript בו תראה סרטונים שיעזרו לך להבין לעומק את השפה. יש לצפות בסרטונים הבאים:

1. introduction – סרטון אחד לא כזה מעניין אך לא מזיק.
2. js foundation
    a. Inside the engine – סרטון מספר 8
    b. writing and optimizing code – סרטון מספר 10
    c. call stack and memory heap – סרטון מספר 12
    d. garbage collection – סרטון מספר 14
    e. Memory leaks – סרטון מספר 15
    f. single threaded – סרטון מספר 16
    g. exrecise issue with single thread -סרטון מספר 17
    h. javascript runtime – סרטון מספר 18
    i. node.js – סרטון מספר 19
    j. Section review – סרטון מספר 21 (דוגמאות טובות)
3. js foundation 2
    a. Execution context – סרטון מספר 2
    b. hoisting stack – סרטון מספר 4
    c. Exercise hoisting 2 – סרטון מספר 6
    d. Exrecise js is weird – סרטון מספר 13 (לא חובה)
    e. function scope vs block scope – סרטון מספר 14 (לא חובה)
    f. this keyword – סרטון מספר 18
    g. exrecise dynamic scope vs lexical scope – סרטון מספר 19
    h. bind () applay ()call() – סרטון מספר 20
    i. bind() and currying – סרטון מספר 22
4. types in js
    a. 7, 10, 11
5. The two pillars closure

a. functions are objects – סרטון מספר 2
b. first class citizens – סרטון מספר 3 (לא חובה)
c. extra bits functions – סרטון מספר 4
d. higher order functions – סרטון מספר 5
e. exrecise higher order functions – סרטון מספר 6
f. closures – סרטון מספר 7
g. exrecise closures – סרטון ממספר 8 (לא חובה)
h. closure and memory – סרטון מספר 9
i. closure and encapsulation – סרטון מספר 10
j. exrecise closure 3 – סרטונים 13 ו-14
k. prototype – סרטונים 16,17,18,19
l. exrecise prototype instance – סרטונים 22,23

6. oop
a. factory function – סרטון מספר 4
b. create – סרטון מספר 5
c. classes – סרטון מספר 9
d. 4 ways – סרטון מספר 11

7. fp
a. functional programming intrudaction – סרטון מספר 2
b. pure functions – סרטון מספר 4
c. pure functions 2 – סרטון מספר 5
d. can everything be pure – סרטון מספר 6
e. idempotent – סרטון מספר 7
f. immutabilitty – סרטון מספר 9
g. hof and closure – סרטון מספר 10
h. currying – סרטון מספר 11
i. mci memoization – סרטון מספר 14
j. mci memization – סרטון מספר 15
k. compose and pipe – סרטון מספר 16
l. solution amazon – סרטון מספר 19

Typescript

העמקה בדקומנטציה של TS, קריאה על סוגי typeים קיימים וקריאה על הנושאים הבאים:

- Union
- Generic
- Intersection
- Keyof
- Typeof
- Partial
- Required
- Readonly
- Record
- Pick
- Omit

Solid ו-cleancode

אנחנו בצוות אובסיביים לסטנדרטים גבוהים ולא מתפשרים על כלום. אז לפני שתתחיל לקבל כאפות ב-merge requestים, אנחנו רוצים שתבין למה זה כזה חשוב לנו.

תקרא את הכתבות הבאות:

- Clean code
- SOLID

ואם אתה גבר מלך ובא לך להיות שפיץ עולם תעיף מבט על הסרטונים הבאים:

- Clean code: the next chapter by victor rentea
- Uncle bob SOLID principles
- Clean code – uncle bob

Design patterns:

תקרא את הכתבה הבאה בשביל להכיר את הרעיון של design patterns ותקרא באתר refactoring guru על הpatternים הבאים:

- Creational
  - Factory ו-abstract factory
  - Singleton
- Structural
  - Adapter
  - Bridge
  - Composite
  - Flyweight
  - Proxy
- Behavioral
  - Observer
  - State
  - Strategy
  - Template method

צד שרת וDB

מונגו דיבי

תקרא על כל הנושאים הבאים

- Mongodb overview
- CRUD in mongo (הכנסה, מחיקה, עדכון ושליפת מידע)
- Aggregation (pipeline)
- Data modeling
- Indexing

אחרי שקראת על כל הנושאים הבאים ואתה מלך המונגו הגיע הזמן לאפיין את הישויות של המערכת שלך

תבנה erd של הישויות של המערכת שלך לפי התיאור של הפרוייקט והפיגמה – אם אין לך כיוון אז תקרא כאן איך מציירים erd א-רלציוני.

NestJS

טכנולוגיית צד שרת מבוססת express/fastify. תוכל למצוא המון מידע בדקומנטציה שלהם (היא מפורטת בטירוף)

תקרא על הנושאים הבאים

- הבדלים בין nest לבין express.
- Controller
- Service
- Module
- Exception filter
- Middlewares/Pipes/Guards/Interceptors
- Lifecycle events
- Request lifecycle
- Tests – integration/unit/e2e

דגשים לכתיבת צד שרת:

- שימוש מעמיק ב Typescript כולל תכנון נכון של האובייקטים שידרשו גם לצד הלקוח

- שימוש בvalidationPipe בכל Route של המערכת
- מימוש Middleware של לוגים ,תיעוד של כל פעולה שמתבצעת וסיווג ה-log לפי קטגוריה: warn, info, error.
- שמירה בקנאות על השכבות ומימוש כל לוגיקה במקום ההגיוני שלה (יכול לקרוא ברפרוף על the three layered architecture)
- יצירת סביבת פיתוח ומשתני סביבה בקבצים רלוונטיים
- תפיסה נכונה של טיפול בשגיאות
- ניהול נכון של הקוד בgit
- סטנדרט קוד של השמחות לרבות:
  - שמירת constants גלובליים במקום רלוונטי
  - קובץ utils עם פונקציות עזר גלובאליות
  - אין שום שימוש במערכת ב-type any, @tsIgnore, ו-as (אלא אם אין פתרון אחר ואז צריך לתת הסבר טוב מאוד)
  - השמת דגש על programming functional
  - שמות משתנים רלוונטים

צד לקוח אנדרואיד

יאללה אנחנו ביישורת האחרונה איזה טירוף – כל מה שנשאר זה לכתוב צד לקוח פסיכוטי שיתן למשרתים באבן להתחרפן במכון :)

אז רגע לפני שנשתגע על הריפוזיטורי צריך לוודא שאנחנו מכירים את הנושאים הבאים:

- Android lifecycle
- MVVM architecture

את הלקוח נפתח בארכיטקטורת MVVM ונשתמש ב-activity אחד (לעוד מידע). בנוסף יש כמה טכנולוגיות שנשתמש בהן (ודברים לקרוא):

- Kotlin
  - Kotlin overview
  - Basics
  - Type checks and casts
  - Control flow
  - Classes and objects
    - Interfaces
    - Extension function
    - Data class
    - Sealed class
    - Delegation properties
  - Functions
  - Null safety
  - Equality
  - This expression
  - Coroutines
  - Scope functions
- Jetpack compose
  - UI architecture
  - Themeing

- Performance
- Style guidelines
- תשתמש בטאב ה-components בשביל הקומפוננטות במערכת.

- Koin
  - אתחול של האפליקציה
  - סוגי injection שונים: ViewModel, Single, Factory
  - להכיר את ההבדלים בין inject ל-get
  - חלוקה למודולים קטנים

סיכום החפיפה

לאורך החפיפה תעבור סקרי קוד גדולים המסכמים חלקים גדולים במערכת תיתן בראש, זה הזמן ללמוד את היסודות כמו שצריך כדי שבהמשך כולם יוכלו לפתח בכיף.

שיהיה בהצלחה!

צוות אלפא (;