תרגול סיום לקורס Vue הצבאי

הקדמה:  
בשלב זה אתם אמורים להבין פחות או יותר קוד ב-JavaScript ואת העקרונות בהם Vue.JS 2.0 משתמש.  
התרגיל הזה הינו תרגיל מסכם ובו אנחנו נבנה שלב אחר שלב אפליקציית whatZapp מורכבת שתכלול את מרבית החומר הנלמד בקורס.

אנחנו הולכים ליצור את האפליקציה בשיטת up-down, משמע, אנחנו הולכים ליצור את ה-components מהחלק העליון ביותר - הרקע לאחרון - פונקציונאליות

יש לדאוג שיש לכם Node.JS על המחשב, כמו כן, VSCode או כל IDE אחר.

ניתן למצוא את כל הקוד לפרויקט כאן:  
<https://github.com/Natanhel/WhatZapp/>

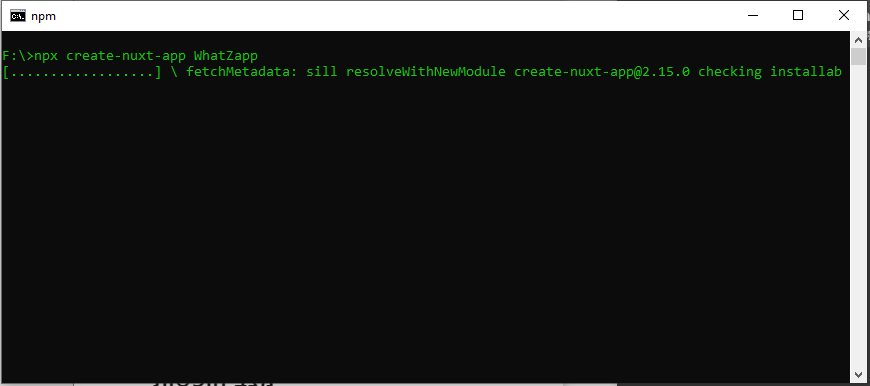
וגירסא שלו על שרת:

<https://whatzapp-clone.herokuapp.com/>

בהצלחה!

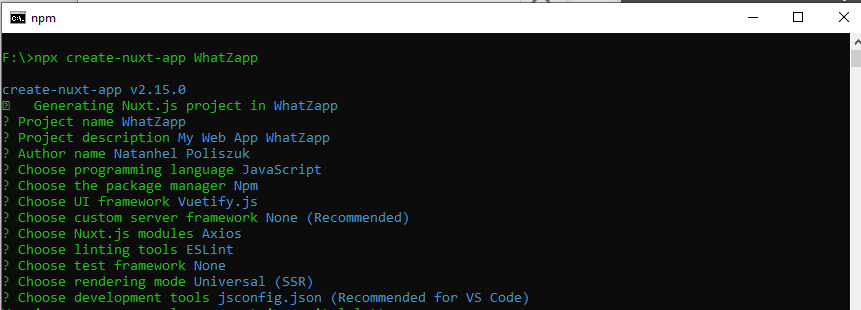
שלב 1  
התקנת Nuxt

בשרת הפקודה יש להגיע למיקום שבו ברצונכם לשמור את הפרויקט החדש באמצעות הפקודה cd בטרמינל.  
כדי למנוע באגים בעתיד: יש לשים את התיקייה **בנתיב קצר** ועל הנתיב להיות בעל **שם באנגלית.**



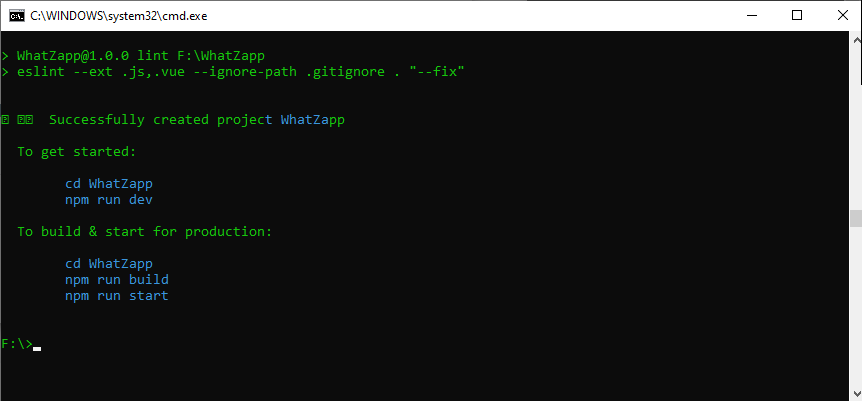
**שימו לב** הפקודה המתקינה את nuxt היא NPX ולא NPM.

יש להתקין את הפרויקט כפי שלמדתם בחלק Scaling With Nuxt.JS עם ההגדרות המתאימות לכם.  
ההגדרות שלי:

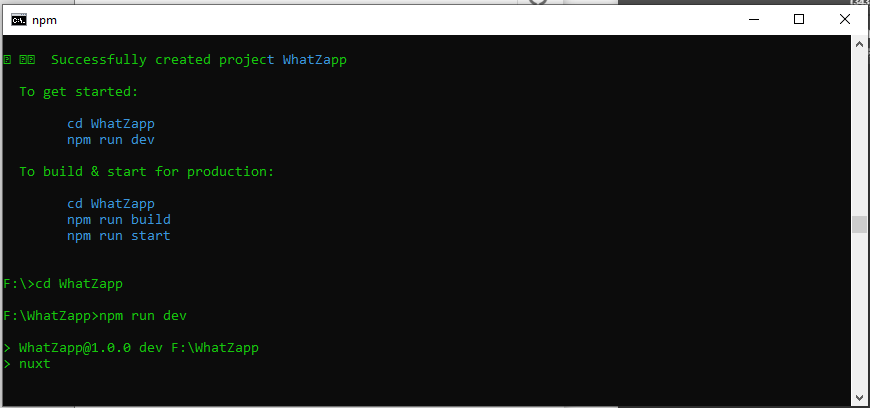


שימו לב שאתם רוצים לכלול את ההגדרות הבאות: Vuetify.Js, Axios, NPM

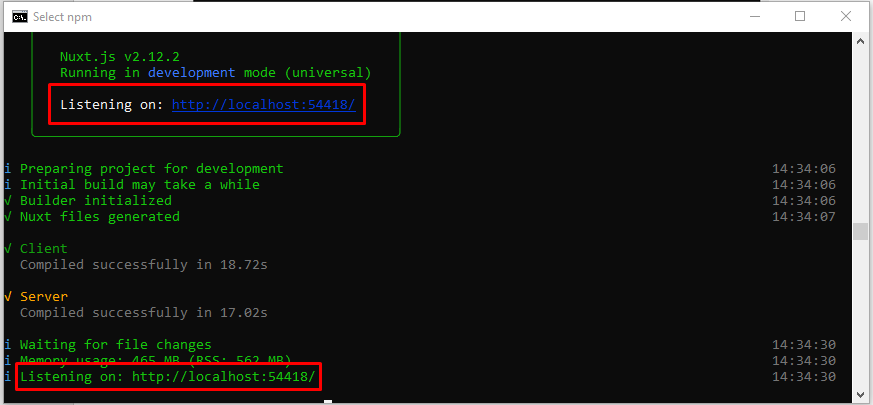
התקנו את הפרויקט בהצלחה!



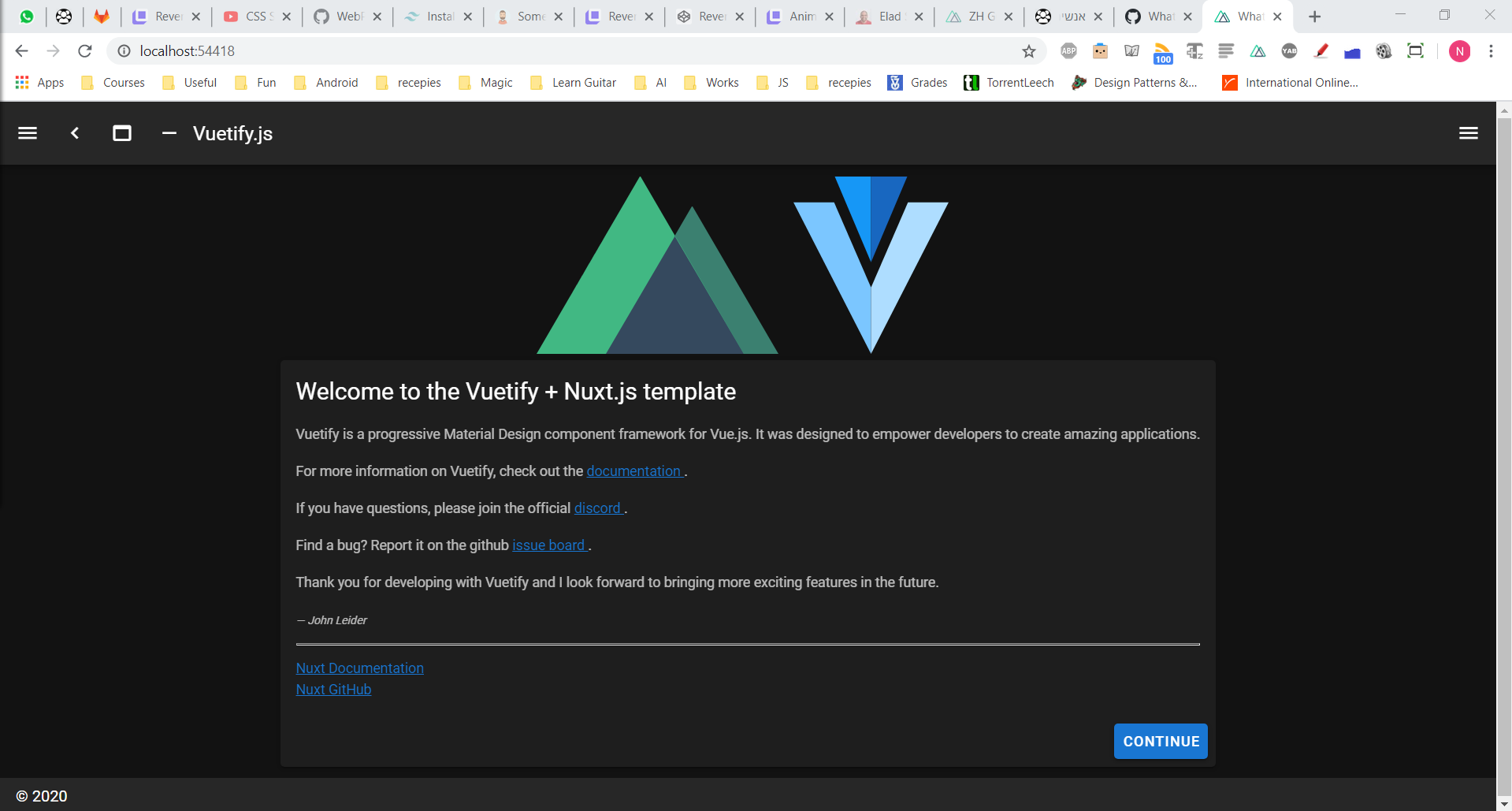
ניכנס לתיקייה ונריץ את השרת בצורה לוקאלית עם הסקריפט dev המוגדר באופן אוטומטי בקובץ package.json שלכם בזמן שהתקנו את nuxt.



כעת פתחו את chrome או כל דפדפן אחר והכניסו לשורת ה-URL את הנתיב לשרת

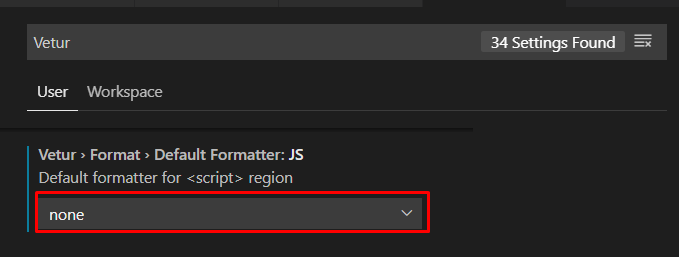


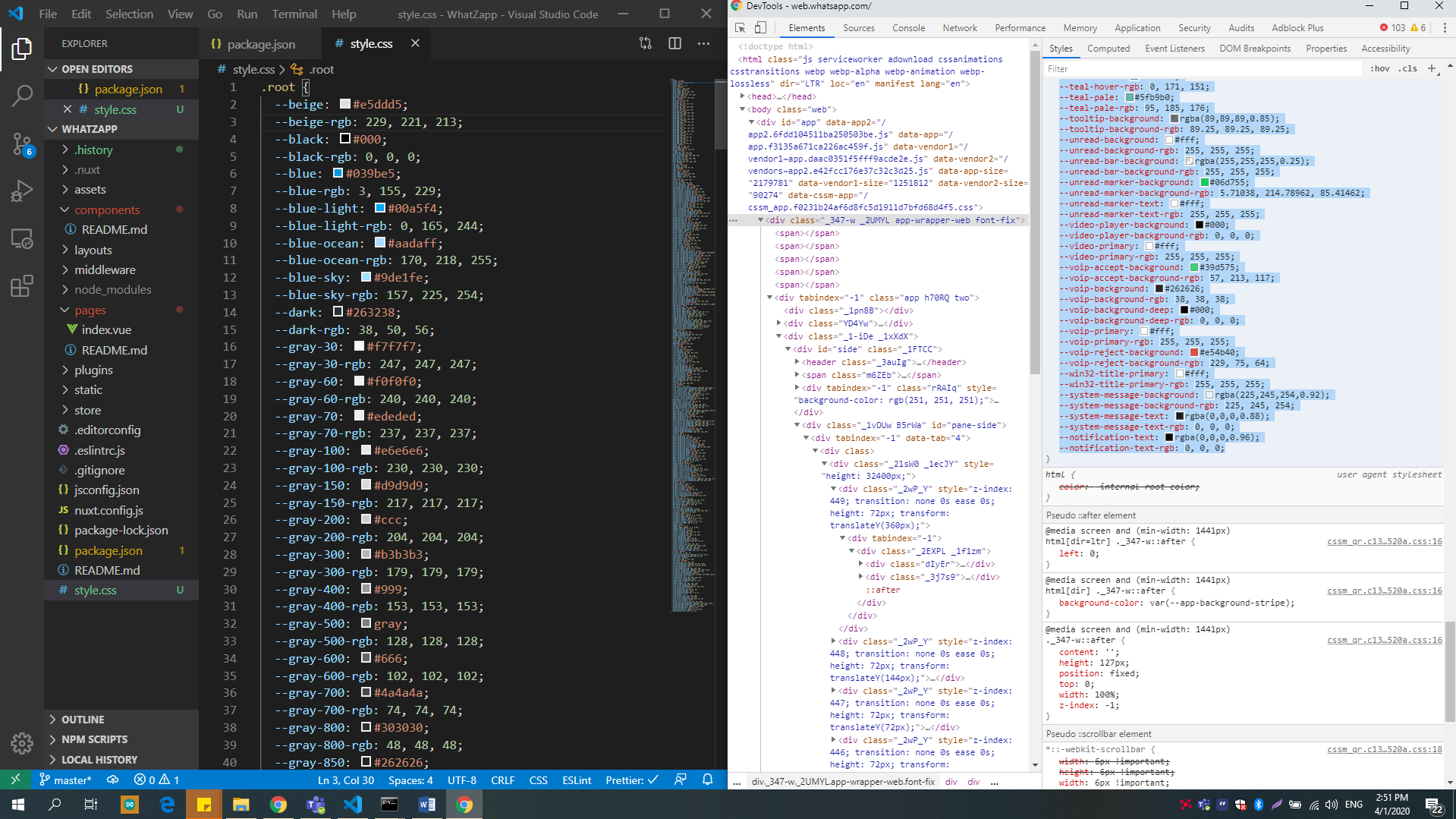
התוצאה:

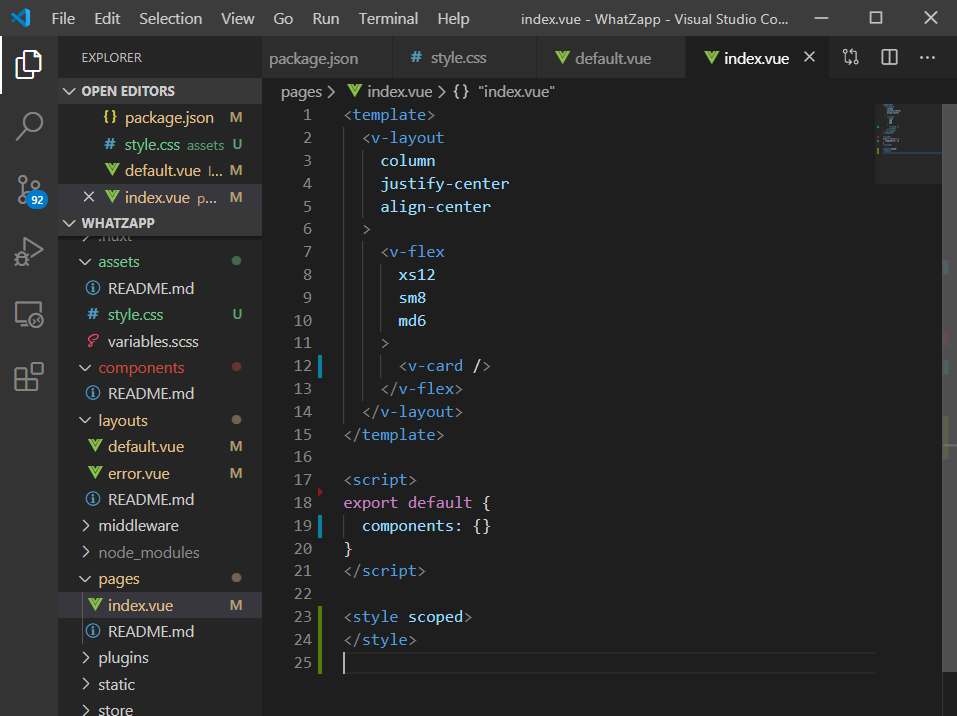


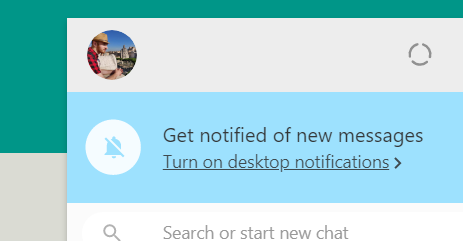
מחקו את ה-Components הבאים: Logo.vue, VuetifyLogo.vue וטפלו בכל ה-dependencies (שלא יהיו שגיאות בפרויקט)

מחקו את ה-Pages הבאים: inspire.vue וטפלו בכל ה-dependencies (שלא יהיו שגיאות בפרויקט)

שימו לב למשהו מעניין, יש באג ב-Vetur שגורם להתנגשות בין ESLint ל-Prettier ולכן עלינו לבטל את התיקון שגיאות של Vetur באמצעות Prettier. לחצו Ctrl + , על מנת לפתוח את האפשרויות של VSCode, חפשו Vetur בשורת החיפוש ושנו את ההגדרות  
  
שימו לב שכעת בזמן שאתם עושים שמירה לא נוסף לכם ';' בסוף שורה וה-ESLint לא יצעק על זה ☺

כעת ברצוננו להתאים את הצבעים המתאימים ל-whatsapp, שלפו מהאתר המקורי את צבעי הרקע באמצעות הכלי Ctrl + Shift + C ושמרו אותם בקובץ css נפרד.  
ניתן להזיז את הקובץ לתיקייה assets לצורך נוחות.

מחקו את כל קובץ ה-index.vue בתיקיית page כך שישאר הקוד הבא:  


נחשו איך ניתן לקבל את הרקע של whatsapp באמצעות קובץ ה-css ששמרנו מקודם והתאמת הצבעים לרקע של whatsapp בתוך ה-component בשם default.vue.  
  
התוצאה:  


מומלץ לנסות בעצמכם אך כאן מוגש לכם הפתרון:

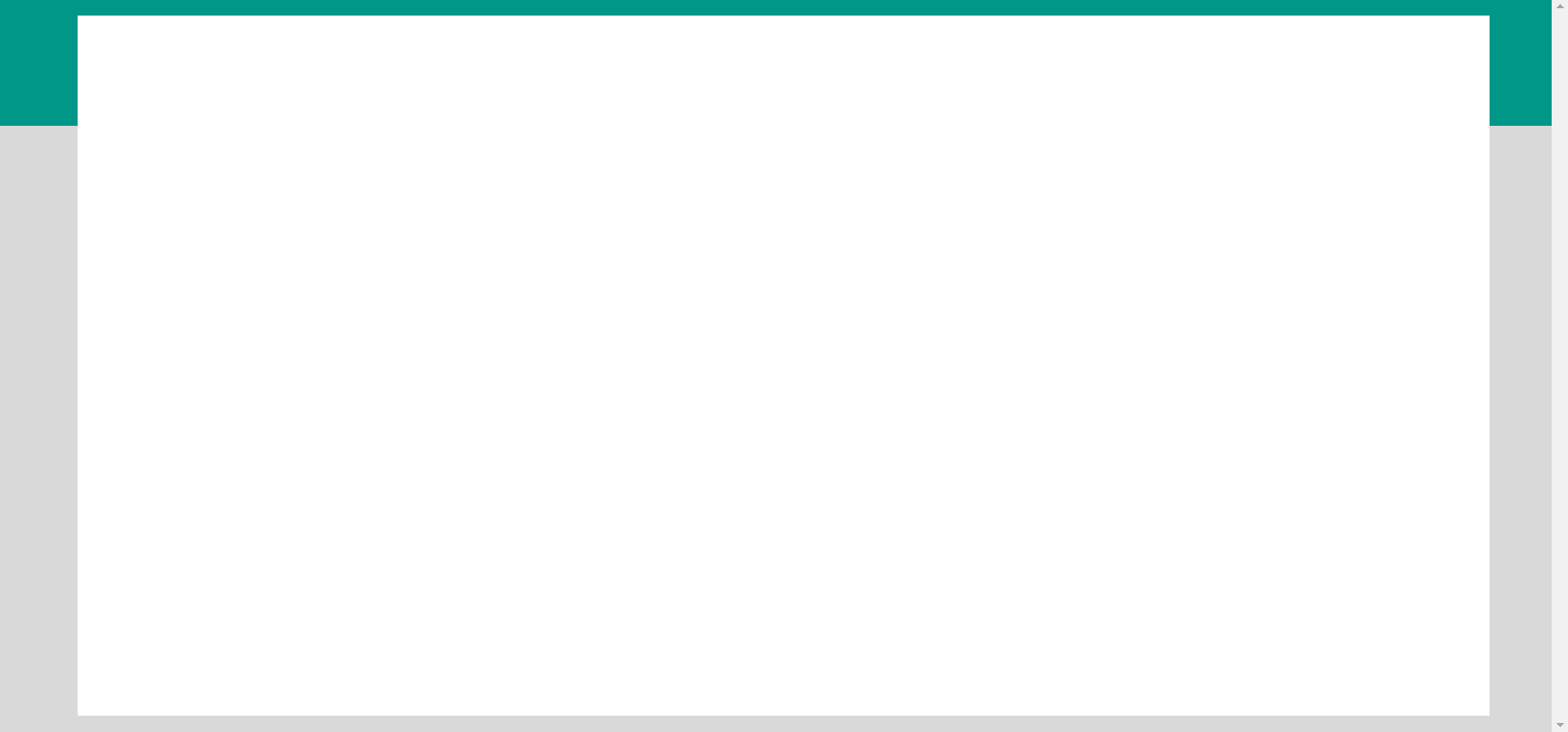
על מנת לראות את הקוד לחצו על התמונה המקובצת כאן:



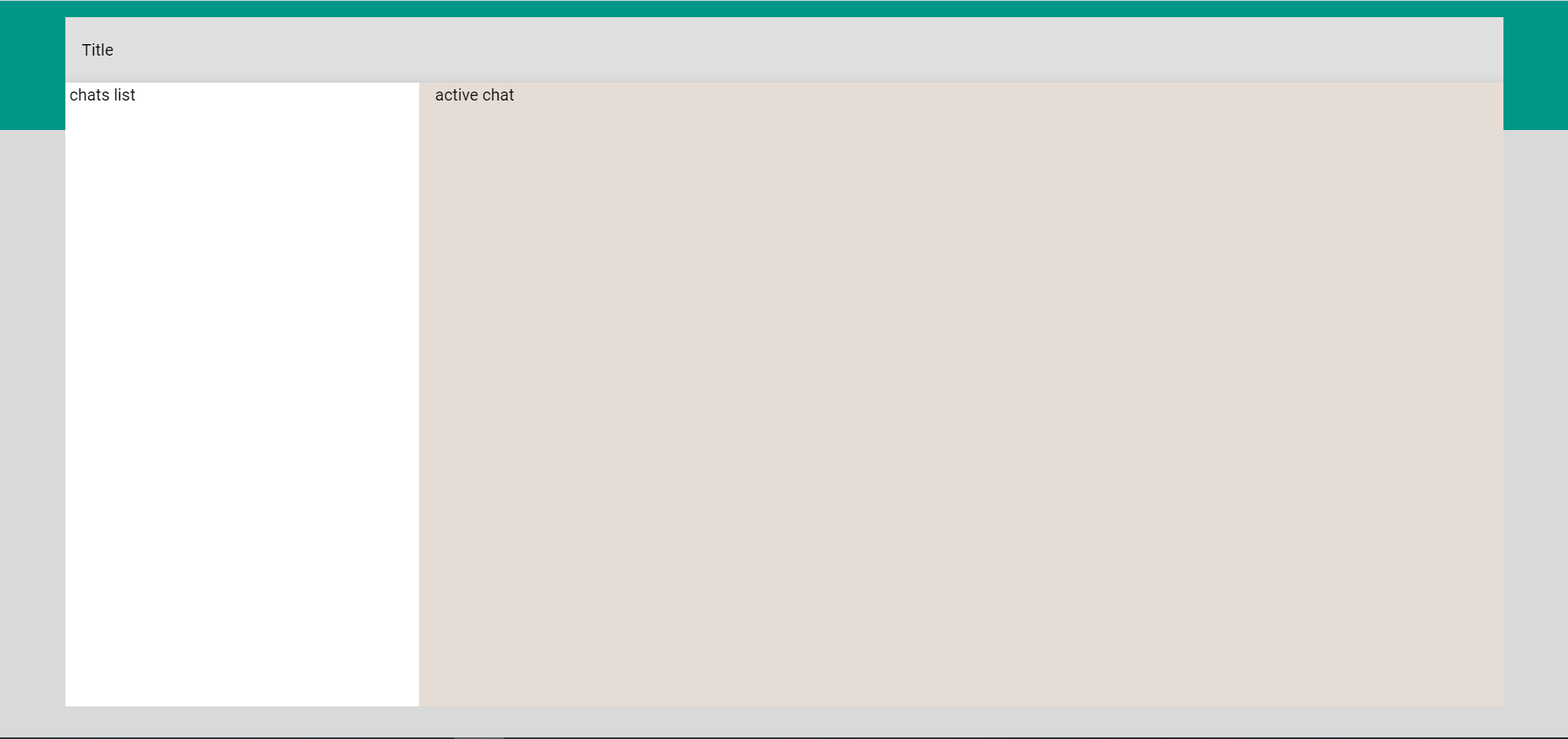
כעת משיצרנו את הרקע ומחקנו את הקוד המיותר ש-nuxt מייצר עבורנו, נרצה ליצור ריבוע שיכיל לנו את האפליקציה הראשית. צרו אחת כזו.

על מנת לראות את הקוד לחצו על התמונה המקובצת כאן:





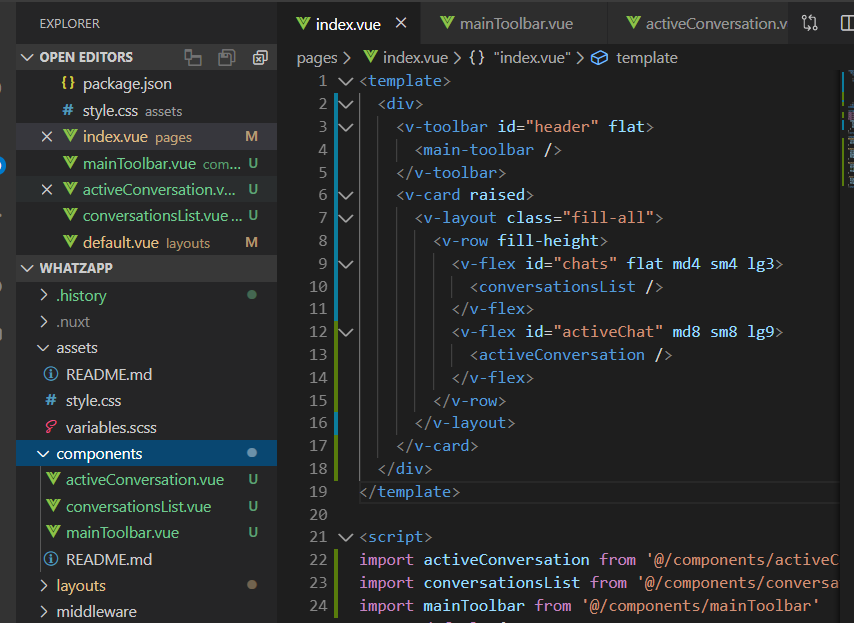
שימו לב שמצד ימין יש פס נגלל, אנחנו רוצים להעלים אותו. מצאו באינטרנט כיצד ובצעו זאת.  
רמז: ה-scrollbar נמצא על התגית html.  
יש כמה דרכים לבצע זאת, הביטו בקוד המקובץ למעלה על מנת לראות כיצד מבצעים זאת על ידי JQuery.

כעת נחלק את המסך לכל התכולות שלו, נסמן את הריבועים שלנו בצבעים שונים:  


הקוד מצורף כאן (לחיצה כפולה):  


כעת נרצה לפתח את ה-toolbar, רשימת השיחות הקיימות והשיחה הנוכחית.  
על מנת לשמור על קוד נקי וקל לתחזוק, נבנה components לכל אחד מהם.  
נשים לב שהשתמשנו בקומפוננטה בשם v-toolbar, אך יש צורך לבנות ל-toolbar שלנו את הקומפוננטה שלו, מכיוון ש-v-toolbar מאפשר לעצב את ה-slot של הקומפוננטה ולכן נוכל להגדיר את ה-toolbar בצורה נוח עם קומפוננטה משלנו

שימו לב שהתיקייה ששמתי בה את ה-components נמצאת באופן טבעי בתיקייה באותו שם.

נקרא לקומפוננטות מעמוד ה-index:  


כעת נוכל להמשיך לעבוד על ה-components בצורה עצמאית ובלתי תלויה ב-components האחרים.

המשימה הבאה שלכם היא למצוא כיצד ניתן לשחזר את ה-toolbar של וואטסאפ ☺

שימו לב, כדי שתוכלו להריץ ב-nuxt את האייקונים של mdi ו-md, עליכם להכניס את השורות קוד הבאות אל תוך קובץ הקונפיגורציה של nuxt: nuxt.config.js תחת החלק vuetify:

defaultAssets: {

font: true,

icons: 'md'

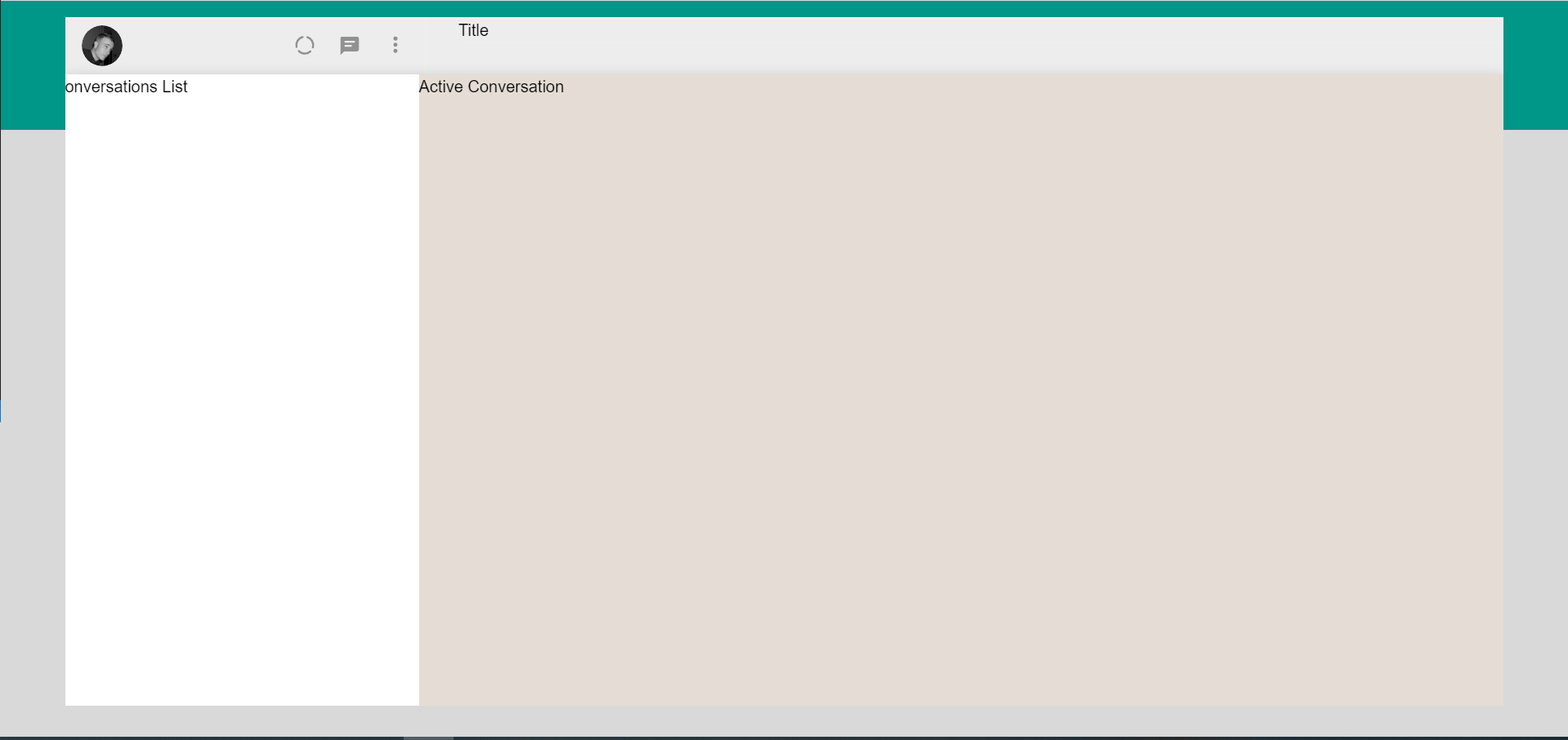
},

icons: {

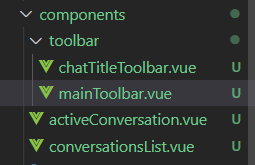
iconfont: 'md',

}

נחלק את העבודה שלנו לחלקים קטנים יותר, עם העבודה תיתכן חלוקה אחרת של תיקיות הפרויקט והקומפוננטות.



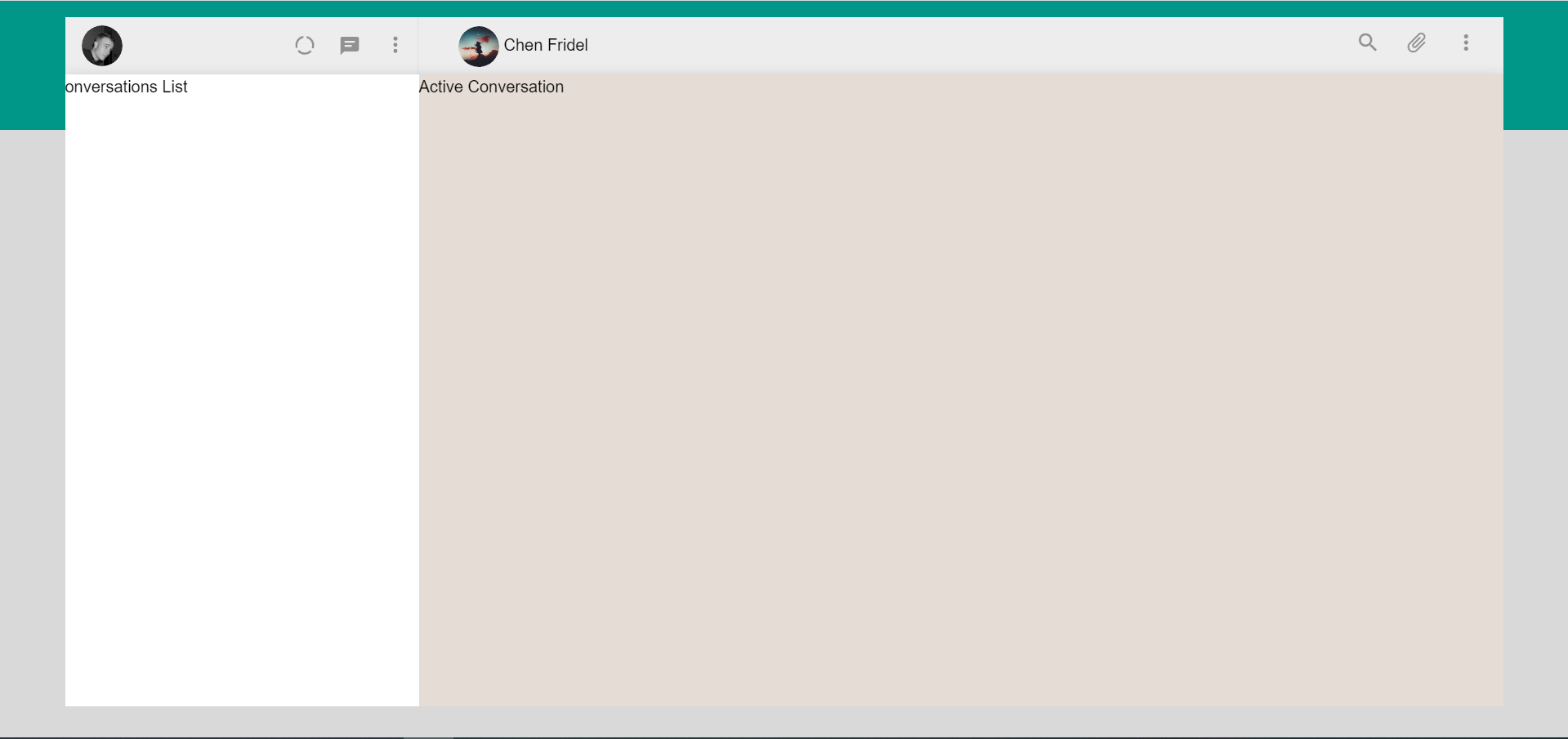
את החלק הראשון שבנינו, יש בו 3 כפתורים, את כל החלק הנ"ל ניתן להכניס לקומפוננטה משל עצמו על מנת שיהיה קוד קריא והקבצים לא יהיו ארוכים מדי.

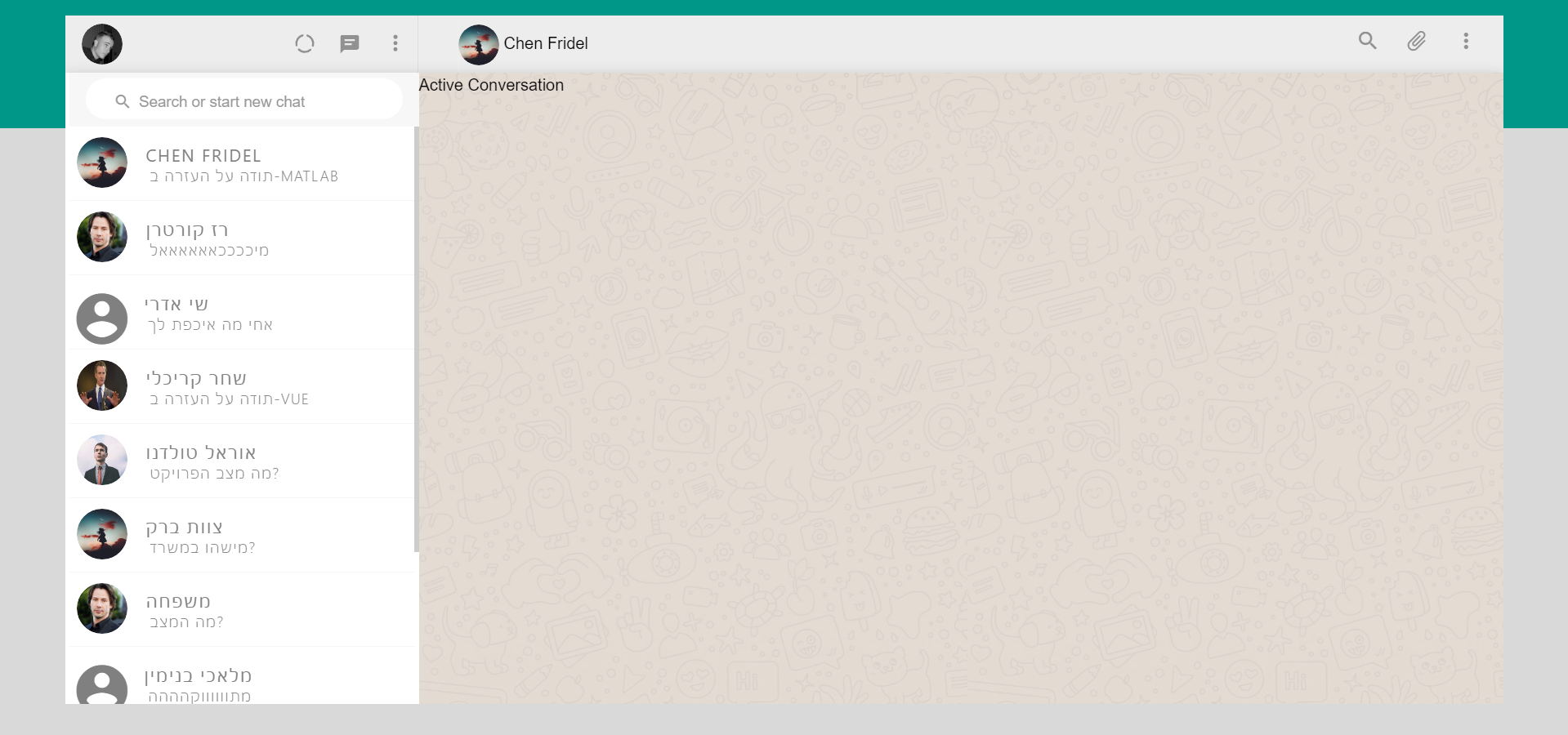


הקוד מצורף כאן (הקלקה כפולה):

|  |  |
| --- | --- |
| mainToolbar.vue | chatTitleToolbar.vue |
|  |  |

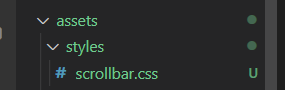
התוצאה:

  
קבצים ארוכים משפיעים על הזמן שלוקח למסך לעלות אך שימו לב שגם ריבוי סקריפטים משפיע על כך ולכן לא נרצה סקריפטים ארוכים. Nuxt פועל בצורה עצלה מבחינת קריאות לסקריפטים ובנוסף לכך שהוא בנוי בצורת ארכיטקטורת SSR (Server Side Rendering) באתר ולכן הסקריפט נטען אחרי שהמסך עלה ורק אם צריך אותו, דבר שמעלה את זמן העלייה של המסך שלנו.

כעת נרצה ליצור קומפוננטה שתכיל את ה-conversation שלנו, נקרא לה conversationNode.vue ונסדר את הקוד בצורה יפה כך שנוכל גם לגלול בצורה דומה לאתר המקורי:  


הקוד של הקומפוננטה נמצא כאן:



כמו כן פתחתי תיקייה חדשה ב-assets על מנת לשמור על קוד CSS משותף בין הפרויקטים:  
  
הקוד ל-scrollbar נמצא בתוך הקובץ ואני מייבא אותו ב-index.vue.  
לפניכם הקוד

\*::-webkit-scrollbar {

  width: 0.3em;

}

\*::-webkit-scrollbar-track {

  background-color: rgba(255, 255, 255, 1);

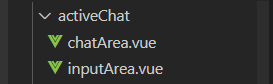
}

\*::-webkit-scrollbar-thumb {

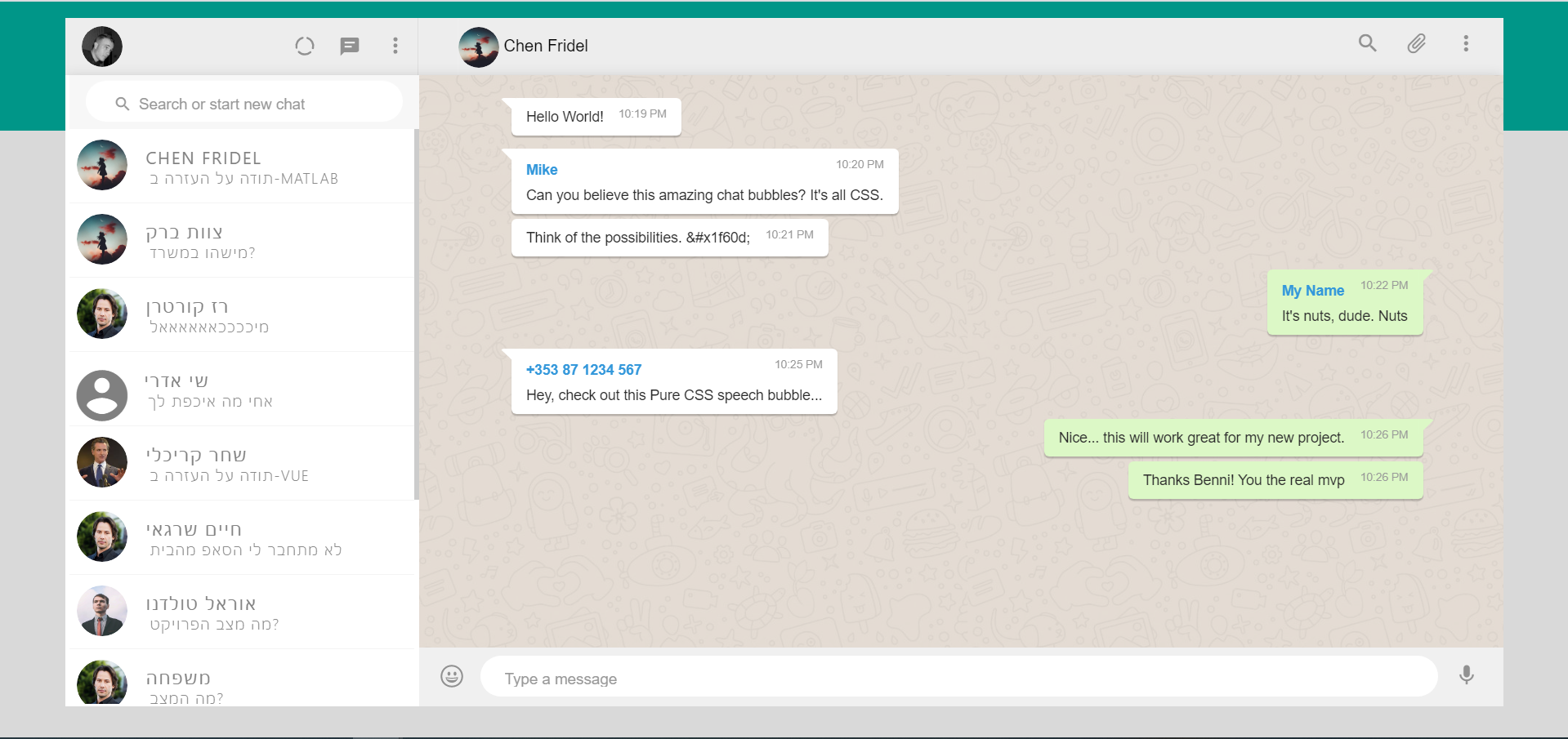
  background-color: rgba(0, 0, 0, 0.2);

}

שימו לב, נוכל לחלק את הקומפוננטה הזו לעוד קומפוננטה כגון search text input.

כעת נעבור לתכנת את אזור הצ'אט עצמו, נוסיף שתי קומפוננטות בתוך תיקייה חדשה, לקומפוננטות נקרא: inputArea.vue ו-chatArea.vue:  


באזור ה-input נשים את הטקסט שנכתב על ידנו ובאזור ה-chat נראה את ההתכתבות שלנו.

בנו את החלוקה המתאימה:  


שימו לב, כרגע אין צורך בבניית אזור צ'אט משתנה לפי לחיצה על הרשימה, נעשה זאת בהמשך.

מצורפים הקבצים:

|  |  |
| --- | --- |
| chatArea.vue | InputArea |
|  |  |

לאחר שבנינו את כל ה-UI, הגיע הזמן להכניס פונקציונאליות למסך.

את המשימות הבאות עליכם לממש ואני לא אספק קוד נוסף, חשבו כיצד לבצע זאת.

חלק מהמימוש תוכלו למצוא ב-github הפרויקט: <https://github.com/Natanhel/WhatZapp>

כמו כן ניתן לראות את תוצאה על שרת <https://whatzapp-clone.herokuapp.com/>

השתמשו ב-Vuex על מנת לאפשר מעבר בין שיחה לשיחה.  
יש ליצור קובץ index.js תחת התיקייה store ו-nuxt ייצר לכם קונפיגורציה מתאימה שתקשר את הקובץ ל-store של Vuex.

שימו העלאה אוטומטית של הצ'אט הראשון ברשימת הצ'אטים במקום הגיוני (רמז: כאשר עולה המסך, אנחנו יכולים לקבל את המידע על הצ'אט הפעיל מרגע עליית רשימת הצ'אטים).

הוסיפו תיבת טקסט דיפולטיבית שתוצג אם הצ'אט ריק.

הוסיפו חיפוש לרשימת הצא'טים שלנו.

שימו לב לרספונסיביות המסך. חקרו על @media query בגוגל וממשו בעצמכם מסך שיגדל ויקטן ויתאים את תוכנו לפי גודל המסך.  
אפשרו הוספת תגובה לצ'אט, שימו לב למי שמגיב ול-type שהוגדרו לכל "chat bubble", תוכלו גם להוסיף תגובה חוזרת אם תרצו והוסיפו השהייה של כחמש שניות.

באמצעות json-server, צרו שיחות על השרת, קבלו אותן באמצעות שאילתות והציגו את הנתונים על המסך