Condo Management System

Code style & Code quality

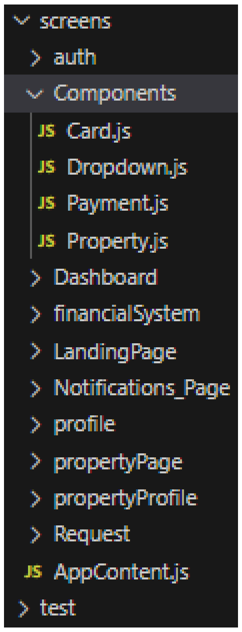
Repository structure

We are implementing a monorepo structure for this project. The backend is separated in its own folder (condo-backend), the frontend into another (condo-management-app), and the pipeline yml files into another (.github).

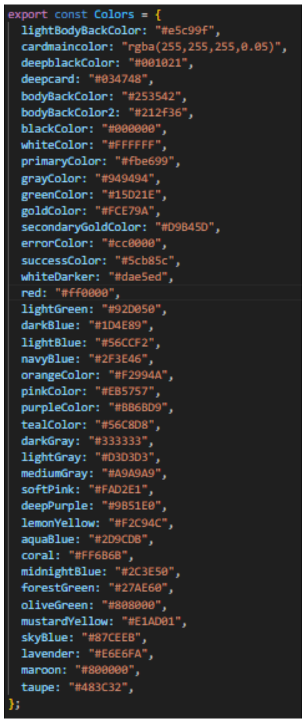
Frontend Coding Styles

Since we are coding the frontend in react, everything is divided into pages and components. Components are reusable pieces of code that are used all throughout the app in all pages. An example is a card component, which simply applies a specific style to the component's children.

All components associated with the same screen are grouped together, allowing for readability and maintainability of the code base. Each component has a specific purpose, allowing for separation of concerns.

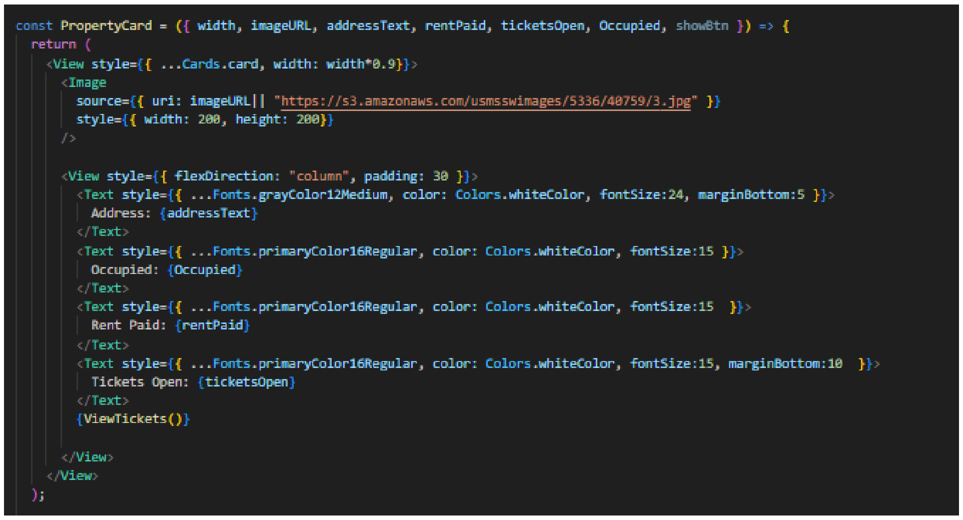


```
screens
  auth
  Components
    Card.js
    Dropdown.js
    Payment.js
    Property.js
  Dashboard
  financialSystem
  LandingPage
  Notifications_Page
  profile
  propertyPage
  propertyProfile
  Request
  AppContent.js
test
```



```
export const Colors = {
  lightBodyBackColor: "#e5c99f",
  cardmaincolor: "rgba(255,255,255,0.05)",
  deepblackColor: "#001021",
  deepcard: "#034748",
  bodyBackColor: "#253542",
  bodyBackColor2: "#212f36",
  blackColor: "#000000",
  whiteColor: "#FFFFFF",
  primaryColor: "#fbe699",
  grayColor: "#949494",
  greenColor: "#15D21E",
  goldColor: "#FCE79A",
  secondaryGoldColor: "#D9B45D",
  errorColor: "#cc0000",
  successColor: "#5cb85c",
  whiteDarker: "#dae5ed",
  red: "#ff0000",
  lightGreen: "#92D050",
  darkBlue: "#1D4E89",
  lightBlue: "#56CCF2",
  navyBlue: "#2F3E46",
  orangeColor: "#F2994A",
  pinkColor: "#EB5757",
  purpleColor: "#BB6BD9",
  tealColor: "#56C8D8",
  darkGray: "#333333",
  lightGray: "#D3D3D3",
  mediumGray: "#A9A9A9",
  softPink: "#FAD2E1",
  deepPurple: "#9B51E0",
  lemonYellow: "#F2C94C",
  aquaBlue: "#2D9CDB",
  coral: "#FF6B6B",
  midnightBlue: "#2C3E50",
  forestGreen: "#27AE60",
  oliveGreen: "#808000",
  mustardYellow: "#E1AD01",
  skyBlue: "#87CEEB",
  lavender: "#E6E6FA",
  maroon: "#800000",
  taupe: "#483C32",
};
```

In order to ensure consistency across all pages, we have a color palette and set of fonts used throughout the app, which can be found in the styles.js file.

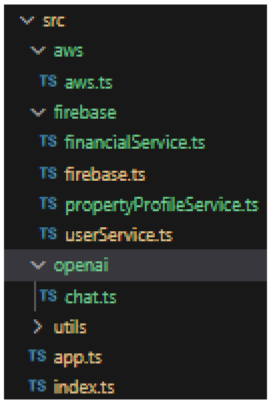


```
const PropertyCard = ({ width, imageURL, addressText, rentPaid, ticketsOpen, Occupied, showBtn }) => {
  return (
    <View style={{ ...Cards.card, width: width*0.9}}>
      <Image
        source={{ uri: imageURL|| "https://s3.amazonaws.com/usmsswimages/5336/40759/3.jpg" }}
        style={{ width: 200, height: 200}}
      />

      <View style={{ flexDirection: "column", padding: 30 }}>
        <Text style={{ ...Fonts.grayColor12Medium, color: Colors.whiteColor, fontSize:24, marginBottom:5 }}>
          Address: {addressText}
        </Text>
        <Text style={{ ...Fonts.primaryColor16Regular, color: Colors.whiteColor, fontSize:15 }}>
          Occupied: {Occupied}
        </Text>
        <Text style={{ ...Fonts.primaryColor16Regular, color: Colors.whiteColor, fontSize:15  }}>
          Rent Paid: {rentPaid}
        </Text>
        <Text style={{ ...Fonts.primaryColor16Regular, color: Colors.whiteColor, fontSize:15, marginBottom:10  }}>
          Tickets Open: {ticketsOpen}
        </Text>
        {ViewTickets()}

      </View>
    </View>
  );
```

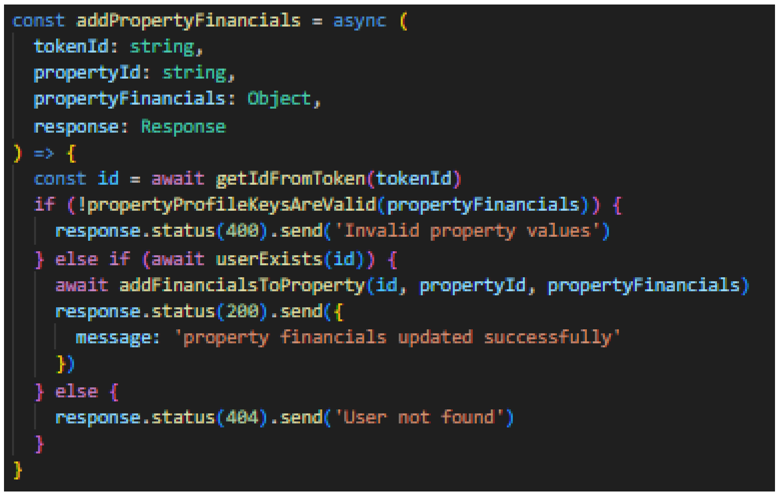
Component, function and variable names follow CamelCase/camelCase styling to help with readability.

Backend Coding Styles

The backend follows a controller/service structure, where the service files are responsible for the bulk of the backend logic and the controllers (in app.ts) simply expose the functions as an endpoint.



```
src
  aws
    aws.ts
  firebase
    financialService.ts
    firebase.ts
    propertyProfileService.ts
    userService.ts
  openai
    chat.ts
  utils
  app.ts
  index.ts
```



```
const addPropertyFinancials = async (
  tokenId: string,
  propertyId: string,
  propertyFinancials: Object,
  response: Response
) => {
  const id = await getIdFromToken(tokenId)
  if (!propertyProfileKeysAreValid(propertyFinancials)) {
    response.status(400).send('Invalid property values')
  } else if (await userExists(id)) {
    await addFinancialsToProperty(id, propertyId, propertyFinancials)
    response.status(200).send({
      message: 'property financials updated successfully'
    })
  } else {
    response.status(404).send('User not found')
  }
}
```

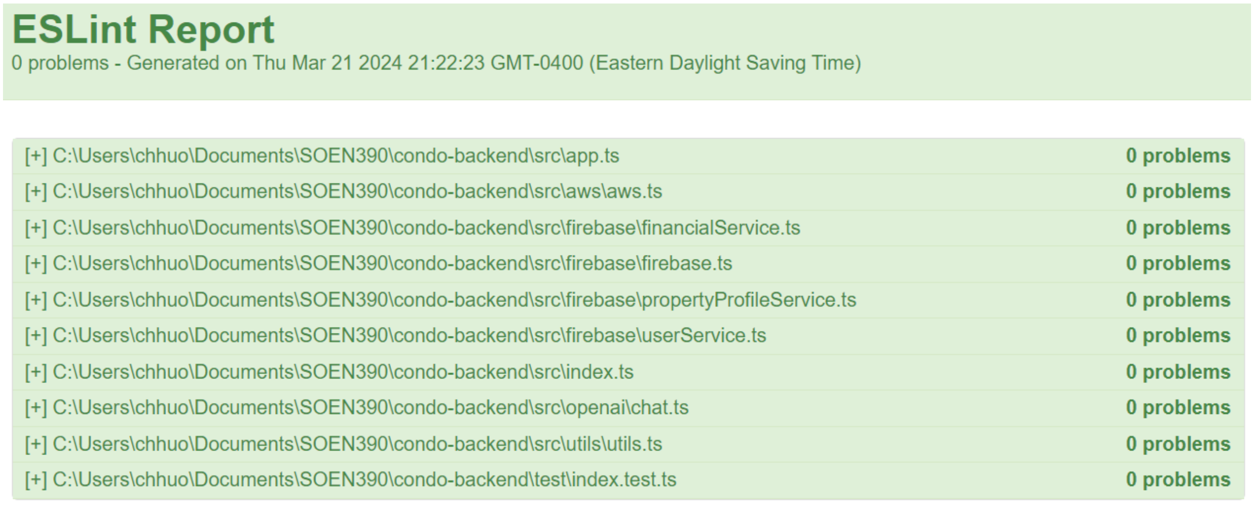
Similarly to the frontend, functions and variables follow CamelCase/camelCase styling to help with readability.

The functions each have a specific purpose and follow the principle of separation of concerns.

Prettier and eslint

We use prettier and eslint during the development to maintain a certain code style and for static analysis. We use the *standard-with-typescript* configuration, which allows us to avoid typical antipatterns and maintain a consistent style.

Here is our ESLint report, showing 0 issues:



```
ESLint Report
0 problems - Generated on Thu Mar 21 2024 21:22:23 GMT-0400 (Eastern Daylight Saving Time)

[+] C:\Users\chhuo\Documents\SOEN390\condo-backend\src\app.ts                                  0 problems
[+] C:\Users\chhuo\Documents\SOEN390\condo-backend\src\aws\aws.ts                              0 problems
[+] C:\Users\chhuo\Documents\SOEN390\condo-backend\src\firebase\financialService.ts            0 problems
[+] C:\Users\chhuo\Documents\SOEN390\condo-backend\src\firebase\firebase.ts                    0 problems
[+] C:\Users\chhuo\Documents\SOEN390\condo-backend\src\firebase\propertyProfileService.ts      0 problems
[+] C:\Users\chhuo\Documents\SOEN390\condo-backend\src\firebase\userService.ts                 0 problems
[+] C:\Users\chhuo\Documents\SOEN390\condo-backend\src\index.ts                                0 problems
[+] C:\Users\chhuo\Documents\SOEN390\condo-backend\src\openai\chat.ts                          0 problems
[+] C:\Users\chhuo\Documents\SOEN390\condo-backend\src\utils\utils.ts                          0 problems
[+] C:\Users\chhuo\Documents\SOEN390\condo-backend\test\index.test.ts                          0 problems
```