Hello.

1. Introduction.

We need a web application that will allow initial users/buyers to pick/buy subscriptions from companies listed on the application by other users. Similarly, the initial users/buyers, who are now existing users/sellers can sell to potential users/buyers and potential users/buyers can buy from existing users/sellers. While buyers can pay with cards, both companies and sellers will get paid to their custom connect accounts with Stripe. For this to happen, all users must have an account with the application and the application will allow users to create a custom connect account that'd be linked to their application account. Since the price of each subscription can fluctuate at any time based on demand and supply, the application will track the price movements for each subscription every 5 seconds. The application uses MongoDB as a database, Stipe as a payment gateway, and next.js as a frontend framework. Also, next.js is required for the backend as well.

1. Register account.

Each application user must have an account and create a custom connect account through the application using the Stripe Custom Connect API.

* 1. Open account

If a user navigates to /account/open, we will show the account registration form. If the user creates an accountAccessKey and clicks on the register button, we will create the user in the database, and redirect to /account/access.

* 1. Access account

If a user enters their accountAccessKey and clicks on the login button, we will redirect to /account.

* 1. Manage account.

If a user navigates to /account, we will show the user’s accountAvatarUrl, accountIsActive, and accountFirstName. Also, we will show the user’s accountCompletionRate based on the number of fields the user has updated from the /account/update form. Basically, there are a total number of 25 form fields and each carries 4 percent. However, the default rate is set to 8 percent because the accountBusinessType and accountBusinessIndustry fields have default values that cannot be changed by the user.

If the user clicks on the profile tab, we will show the user’s default accountAvatarUrl and accountCurrentTitle. If the user makes changes to either of the values, we will update the database. If the user clicks on the settings tab, we will show the user’s accountAccessKey and accountSecretKey. If the user makes changes to the former, we will update the database, but we will not allow the user to update the latter. If the user clicks on the alerts tab, we will show all the user’s subscription transaction details, which will include the accountAlertLogo, accountAlertUnits, accountAlertTicker, accountAlertType, accountAlertStatus, accountAlertPrice, and the accountAlertIsDated of each subscription.

* 1. Update account

If a user navigates to /account/update, we will allow the user to enter/edit/view their accountGovernmentId, accountIdNumber, accountBusinessName, accountBusinessWebsite, accountBusinessAddress, accountBusinessCountry, accountHomeAddress, accountZipCode, accountCityName, and accountStateName. However, we will only allow the user to enter/view their accountFirstName, accountLastName, accountHomeCountry, accountBirthDate, accountBusinessEmail, accountBankCountry, accountBankCurrency, accountIbanNumber, accountNumber, accountRoutingNumber, accountSortCode, accountEmailAddress, and accountPhoneNumber. Similarly, we will not allow the user to enter/edit their accountBusinessType and accountBusinessIndustry.

If the user submits the form, we will calculate the accountCompletionRate of the user and show the rate in the /components/cards/Completion.js file. If the rate is more than 80 percent, we will use the Stripe Connect API to create a custom connect account for the user, pass currently\_due as the value of the collect parameter in our calls to the Stripe Account Links API, accept the Stripe service terms through API, upload the government id of the user to Stripe for verification purposes, save the user information to the database, and enable the user to submit push requests. However, if the user’s rate is less than 80 percent, we will not create a custom connect account for the user and we will not allow the user to submit a push request to sell subscriptions.

If the user enters letters that match a countryCurrency in the database, we will use search-as-you-type to show the matched countryCurrency in the accountCountryCurrency field. Also, if the user enters letters that match a countryCode in the database, we will use search-as-you-type to show the matched countryCode in the accountCountryCode field. If the user enters the US as the countryCode, we will specify card\_payments and transfers as the capabilities, specify full as the service agreement, and specify destination as the transfer\_data value for destination charges purposes. However, if the user enters a non-US, i.e. ES, as the countryCode, we will specify transfers as the capabilities and specify recipient as the service agreement.

If the accountBusinessName field is null, we will use the user’s accountFirstName and accountLastName. If the accountBusinessCountry field is null, we will use the user’s accountHomeCountry. If the accountBusinessAddress field is null, we will use the user’s accountHomeAddress. If the accountBusinessEmail field is null, we will use the user’s accountEmailAddreess.

* 1. Recover account

If a user enters their accountSecretKey and clicks on the continue button. If the key exists in the database, we will redirect to /account/open/\_id. If the user creates a new accountAccessKey and clicks on the done button instead of the register button, we will redirect to /account, and update the user account in the database.

* 1. Whitelist account

If a new user enters their email address on any /[id] route and clicks on the continue button, we will redirect to /account/open so the user can follow the normal account opening process. Similarly, if the user opens an account, we will add the user to the companyIsoSubscribers of the company linked to the /[id] route. If an existing user enters their email address and clicks on the continue button, we will redirect to /account, and add the user to the companyIsoSubscribers of the company linked to the route.

* 1. Exit account

If a user clicks on the logout link in the /components/grids/RightGrid.js file and logs out from their account, we will redirect to /, and change the link to login. Similarly, if the user clicks on the login link and logs in to their account, we will change the link to logout. If the accountIsAdmin of a logged-in user is FALSE and the user attempts to access any /api/ … routes, we will redirect to /account, and redirect logged-out users to /account/access. If the user is logged-in or logged-out, we will show the top 20 companies with the highest companyCapitalization in the /components/menus/TickerMenu.js file.

If the user is not logged-in and attempts to access the /account, /account/update, /companies/list, /companies/ticker, /subscriptions/[id], /plans, or /dashboard routes, we will redirect to /account/access. Similarly, if the user is logged-in and attempts to access the /companies/[id] they did not create, we will redirect to /companies. Also, if the user attempts to directly navigate to /account/open/\_id without providing their accountSecretKey on /account/recover, we will redirect to /account/access.

1. Submit company
   1. List company
   2. Manage company
2. Request subscription
   1. Pick subscription
   2. Push subscription
   3. Pull subscription
   4. Match subscription
      1. Peer same data
      2. Peer higher data
      3. Peer lower data
      4. Peer null data
      5. Peer varied data
      6. Peer maker data
      7. Peer none data
   5. Manage subscription
   6. Track subscription

hdhhdhd