Clone of a Credit offer marketplace

Job headline

We are looking for an experienced Full Stack Spring java developer to clone a simple form filling application consisting of seven pages with a few business rules. The web application should be cloned and front ended roughly adapted to our brand look and feel. It should be written in Spring with java, storing form data to Google Datastore. All frontend to backend interactions should be through rest endpoints.

Deliverables:

Private Github repo consisting of a maven spring boot project. A README file explaining how to build, configure and start application. It should be easy to start and test application from scratch. Google Datastore initial data (combo fields, etc) should be populated on the startup. Ideally it should be a matter of cloning git repo, building maven, configuring google datastore credentials file and run it!

We want a clone of a website "<u>www.bompracredito.com.br</u>". It consists of a simple step by step form filling seven pages with a few business rules. The actual credit rules will be implemented later by us. We just want a working page by page form filling application that connects to the spring boot back end and store data to Google Datastore.

Requirements:

- Spring Boot application, (2.1.3)
- Spring gcp integration (spring-cloud-gcp-starter-data-datastore)
- Apply spring best practices for datastore repository. See
 https://cloud.spring.io/spring-cloud-static/Greenwich.RC1/multi/multi_spring_data_cloud_datastore.html
- Each form being filled should be stored in google datastore, through a rest endpoint.
- The look and feel, overall site "look" and ux components should appear as https://www.acordocerto.com.br/proposta
- The application should be easy to maintain and update. Method names and code can be in English.
- The front end pages should all be in Portuguese, as original website.
- The webservice been invoked might return errors validation errors for specific fields on the page being filled.
- We will use Slack to communicate often.

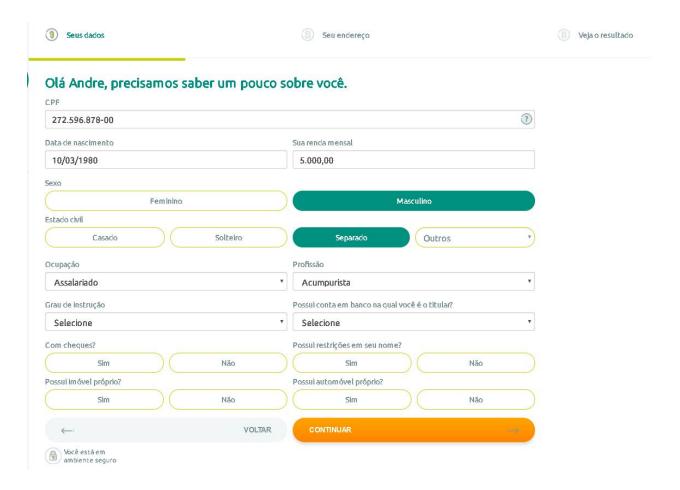
All form filling steps should have a "Voltar" button (go back) that goes back to previous step, all preserving state.

The "Continuar" button that is present in almost all pages should only be enabled when all fields are filled.

Have a look of the entire form filling process in this video: https://www.youtube.com/watch?v=bdClyUanpKA

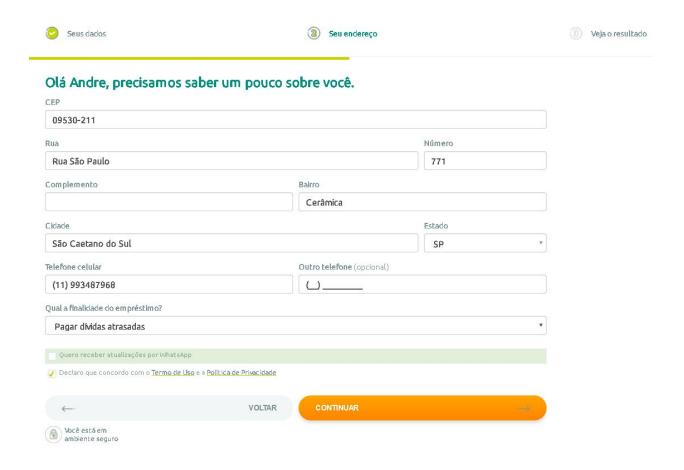
R\$2.000 R\$3.000	R\$5.000 R\$10.000 Outro valor
R\$2.000	(85.000 1610.000 Outlo Valor
Em quantas veze	es você quer pagar?
9х 12х	18x 24x Outro prazo
Nome Completo	
E-mail	
	o Bom Pra Crédito comunicações por email e
E-mail Aceito receber d	o Bom Pra Crédito comunicações por email e CONTINUAR »

For each step the customer goes through, data should be sent to a rest endpoint, and saved to datastore.



All combo data should be fetched from a rest endpoint.

Data for "Profissão" depends on selection of (Ocupação)

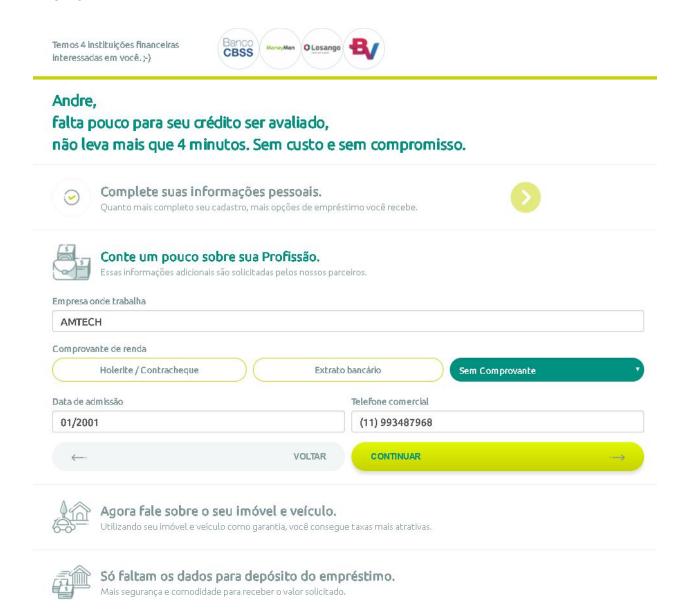


Once "CEP field" is filled, webservice "GetAddress(string cep)" should be called . This webservice should call brazilian adress database webservice: https://correiosapi.apphb.com/ . fill fields "RUA", , "BAIRRO", "CIDADE" and "ESTADO"

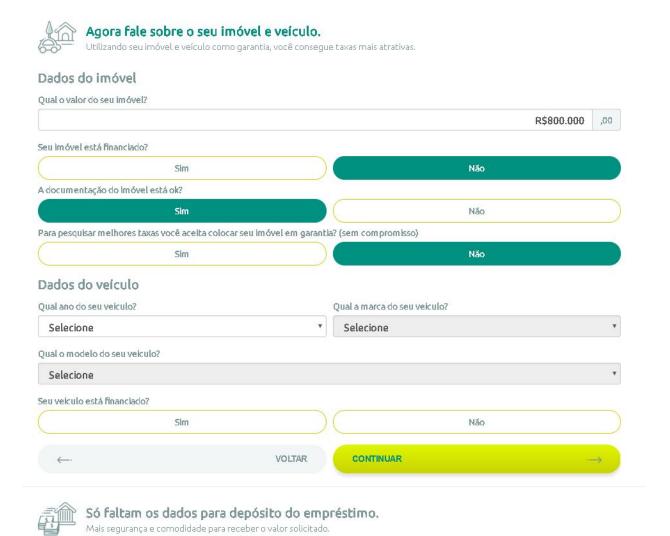
Save results do datastore and user click "Continuar" as usual



Intermediate step. Should call a "pre-process webservice" da stores last page form data and wait for 5 seconds. The transition to the next page



Simple form. Simply save data to datastore



All combos should be fetched from webservice.

Combo "Qual modelo do veículo" depends on "Qual marca do veículo"



Simple form. Just submit it to rest and save to datastore

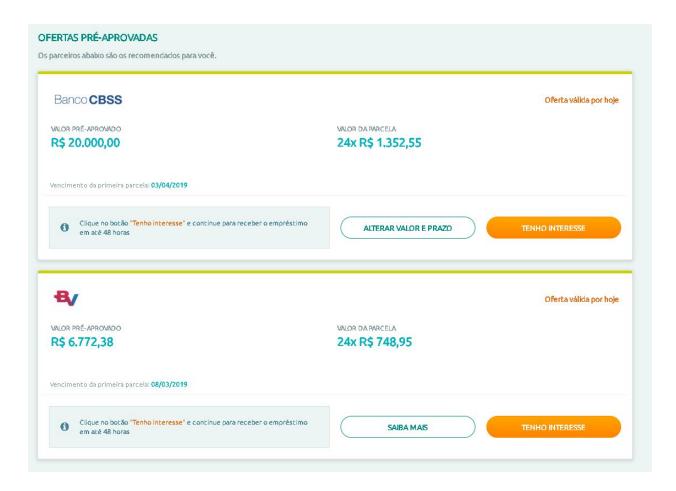


Estamos Buscando as ofertas disponíveis para você!

Mas fique tranquilo, as opções também serão enviadas por SMS, e-mail ou via contato telefônico. Fique atento

Call "process" webservice with last page data wait for response. Show this waiting screen until response is done.

The "process" webservice should fake this response data delay. It should wait for 5 seconds



This the result of "process" webservice. Fake a webservice response data that looks like the page above