

Recruitment System – dokumentacja etapu 2

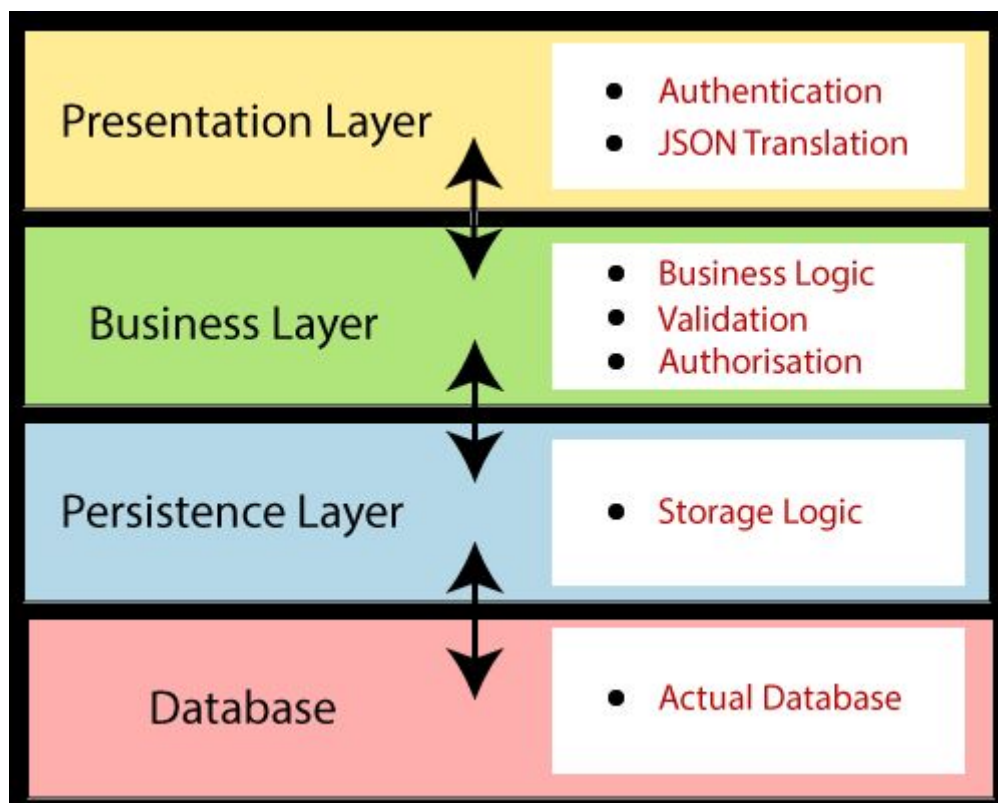
Podsumowanie rozwiązań technologicznych na aktualnym etapie prac:

Axios – klient http dla przeglądarek i NodeJS, oparty o promise'y. Pozwala na formowanie requestów i obsługę odpowiedzi. Axios automatycznie transformuje dane do postaci łańcuchu znaków oraz pozwala na konfigurację requestów za pomocą funkcji dostarczanych przez tzw. interceptors.

Data Transfer Object (DTO) – są to obiekty pełniące rolę pośrednika między warstwą biznesową aplikacji a warstwą widoku. Są one uproszczoną reprezentacją encji znajdujących się w bazie danych, różniące się od nich brakiem pól zbędnych z punktu widzenia potrzeb prezentacji danych na interfejsie użytkownika, oraz brakiem adnotacji z frameworka Hibernate. Dzięki temu, stosując je upraszczamy obsługę przesyłu danych między aplikacją frontendową a backendową, oraz zmniejszamy rozmiar odpowiedzi odsyłanych jako pliki JSON.

Formik – narzędzie służące do tworzenia formularzy oraz ich obsługi (walidacja, obsługa zdarzeń). Głównymi zaletami Formika jest łatwość w dostępie do wartości przechowywanych przez pola, walidacja danych, informowanie o błędach i pomoc w obsłudze zdarzeń.

Layered Architecture – jest to podział aplikacji na warstwy, w którym każda warstwa ma swój precyzyjnie określony zakres obowiązków, oraz zachowana jest zasada jednokierunkowej komunikacji – warstwy niższe nie wywołują warstw wyższych. Architekturę tę przedstawia poniższa grafika:



Aplikacja oparta na takiej architekturze ma spójną i ujednoliconą strukturę, co ułatwia zespołowi programistów współpracę ze sobą – zachowanie takiego pozwala rozwój aplikacji w formie powtarzalnych kroków, zatem unikamy przez to konfliktów, które mogłyby wynikać z różnic w sposobie implementacji funkcjonalności między osobami w zespole. W naszej aplikacji architektura ta jest dobrze widoczna dzięki adnotacjom dostarczonym przez framework Spring: warstwę prezentacji stanowią kontrolery oznaczone jako `@Controller` lub `@RestController`. Warstwę biznesową stanowią komponenty z adnotacją `@Service`. Warstwę dostępu do danych stanowią komponenty oznaczone przez `@Repository`. W ten sposób

PostgreSQL – system zarządzania relacyjnymi bazami danych. Nasz wybór oparty był na powszechności tego narzędzia, oraz jego prostej i intuicyjnej obsłudze. W trakcie realizacji projektu będziemy wykorzystywać Postre w wersji 11.

Session based authentication – podstawowa metoda uwierzytelniania, polegająca na utworzeniu przez serwer sesji dla aktualnie logującego się użytkownika i odesłaniu mu odpowiedzi w postaci tzw. cookie, który jest wysyłany za każdym razem, gdy użytkownik wyśle żądanie dostępu do zasobów chronionych serwera. Pomimo wad związanych z niskim poziomem bezpieczeństwa, zdecydowaliśmy się na wybór tej metody na obecnym etapie prac, ze względu na większą, w porównaniu do innych rozwiązań, prostotę implementacji.

Spring Boot – framework będący rozwinięciem frameworka Spring. Jego rolą jest pomoc programiście w konfiguracji projektu, w myśl zasady **convention over configuration**.

Pomoc ta polega min. na dostarczeniu wbudowanego do projektu serwera Tomcat czy też ułatwieniu zarządzania zależnościami oraz strukturą projektu. Oparcie naszego projektu o Spring Boot pozwala nam ograniczyć ilość koniecznych konfiguracji do projektu, oraz łatwiej obsługiwać go w trakcie jego rozwoju.