

Technologie wykorzystane w projekcie:

- RESTful,
- BootStrap,
- HTML 5.0,
- JavaScript,
- MongoDB,
- IndexedDB.

Dokumentacja:

- opisy skryptów JavaScript zostały zamieszczone w kodzie źródłowym

<http://pascal.fis.agh.edu.pl/~8sudol/projekt2/rest.js>,

- styl RESTful:

- *api.php*:

- *_login* – metoda realizująca logowanie do serwisu. Metoda ta wywołuje metodę *Login* znajdującą się w *mongo.php*, która sprawdza, czy użytkownik o podanym loginie i hasle istnieją w bazie, w przypadku powodzenia wywoływana jest kolejna metoda *NewSession*, która tworzy nową sesję, zwracając przeglądarce informacje o powodzeniu operacji oraz wartość nowego tokenu sesji, w przeciwnym wypadku w przeglądarce wyświetlany jest odpowiedni komunikat o błędzie.

- *_register* – metoda realizująca rejestrację użytkownika. Metoda ta wywołuje metodę *RegisterAdd* znajdującą się w *mongo.php*, w przypadku powodzenia przeglądarce zwracana jest informacja o powodzeniu operacji, w przeciwnym wypadku w przeglądarce wyświetlany jest odpowiedni komunikat o błędzie.

- *_add* – metoda realizująca dodawanie nowych rekordów do bazy danych po stronie serwera. Metoda ta wywołuje metodę *AnswerAdd* znajdującą się w *mongo.php*, w przypadku powodzenia przeglądarce zwracana jest informacja o powodzeniu operacji, w przeciwnym wypadku w przeglądarce wyświetlany jest odpowiedni komunikat o błędzie.

- *_read* – metoda realizująca odczytywanie rekordów z bazy danych po stronie serwera. Metoda ta wywołuje metodę *Read* znajdującą się w *mongo.php*, w przypadku powodzenia przeglądarce zwracana jest informacja o powodzeniu operacji, w przeciwnym wypadku w przeglądarce wyświetlany jest odpowiedni komunikat o błędzie.

- *_sessionCheck* – metoda sprawdzająca, czy dla aktualnie zalogowanego użytkownika trwa jeszcze sesja. Metoda ta wywołuje metodę *SessionCheck* znajdującą się w *mongo.php*, która sprawdza, czy podany token sesji znajduje się w bazie oraz czy jego żywotność nie przekroczyła 10 minut, a następnie wywoływana jest metoda *GetSessionUser*, która zwraca login użytkownika przypisany do omawianego tokena sesji. W przypadku powodzenia przeglądarce zwracana jest informacja o powodzeniu operacji wraz loginem użytkownika, w przeciwnym wypadku w przeglądarce wyświetlany jest odpowiedni komunikat o błędzie.

- *_logout* – metoda realizująca wylogowywanie użytkownika z serwisu. Metoda ta wywołuje metodę *DestroySession* znajdującą się w *mongo.php*, która niszczy token ówczesnie zalogowanego użytkownika. W przypadku powodzenia przeglądarce zwracana jest informacja o powodzeniu operacji, w przeciwnym wypadku w przeglądarce wyświetlany jest odpowiedni komunikat o błędzie.

- *mongo.php*:

- *Read* – funkcja wysyłająca zapytanie do kolekcji *survey* w celu odczytania wszystkich rekordów z bazy.

- *RegisterAdd* – funkcja realizująca dodanie nowego użytkownika do bazy. W tym celu najpierw wysyłane jest zapytanie do kolekcji *users*, czy przypadkiem nie ma już takiego rekordu, a jeżeli nie, użytkownik zostaje pomyślnie dodany.

- *Login* – funkcja realizująca logowanie użytkownika do serwisu. W tym celu wysyłane jest zapytanie do kolekcji *users*, czy istnieje rekord o podanym loginie i hasle. Jeżeli zwracana odpowiedź nie jest pusta, użytkownik zostaje pomyślnie zalogowany.

- *AnswerAdd* – funkcja realizująca dodanie nowych odpowiedzi pochodzących z wypełnionej wcześniej ankiety. W tym celu wysyłany jest prosty insert do kolekcji *survey* zawierający odpowiedzi, nazwę użytkownika oraz datę dodania.

- *SessionCheck* – funkcja sprawdzająca, czy podany token sesji jest aktualny. W tym celu wysyłane jest zapytanie do kolekcji *sessions*, aby otrzymać informacje o aktualnej sesji, a następnie dla otrzymanej odpowiedzi sprawdzane jest, czy czas od rozpoczęcia się owej sesji do teraz nie przekroczył 10 minut – jeżeli nie, zwracana jest wartość logiczna *true*, w przeciwnym wypadku owy token usuwany jest z bazy, a dalej zwracana wartość logiczna *false*.

- *GetSessionUser* – funkcja zwracająca login użytkownika dla konkretnej wartości tokenu sesji. W tym celu wysyłane jest zapytanie do kolekcji *sessions*, z której to odpowiedzi wyodrębniana jest nazwa użytkownika.

- *NewSession* – funkcja zwracająca token nowej sesji dla podanego użytkownika. W tym celu tworzony jest nowy token przy pomocy funkcji *uniqid()*, a następnie wykonywany jest insert do kolekcji *sessions* zawierający wyżej wymieniony token, login użytkownika oraz aktualną datę.

- *DestroySession* – funkcja usuwająca zadany token sesji z kolekcji *sessions*.

Przechowywanie danych:

- dla bazy danych po stronie serwera występują 3 kolekcje:

- *users* – zawiera id, login oraz hasło użytkownika,

- *sessions* – zawiera id, wartość tokenu sesji, login użytkownika, który jest aktualnie zalogowany oraz datę rozpoczęcia sesji,

- *survey* – zawiera id, datę dodania rekordu, login użytkownika, który owy rekord dodał oraz jego odpowiedzi pochodzące z ankiety.

- dla lokalnej bazy danych znajdującej się w przeglądarce występuje kolekcja *surveyLocal*, która zawiera:

- *survey* – zawiera autoinkrementujące się id, podane odpowiedzi do ankiety oraz datę dodania rekordu.