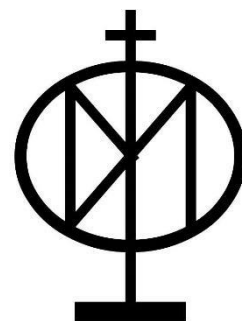




Софийски университет  
„Свети Климент Охридски“  
Факултет по математика и  
информатика

---



**Проект по Fullstack Application**  
**Разработка на клиент-сървър (fullstack)**  
**приложения с Node.js + Express.js +**  
**React.js**

**Тема:**

***Hirundo***

Светимир Игнатов 61726

Емине Башева 61729

Ръководител:

(ас. Т.Илиев)

## Съдържание

<b>1. Идея на проекта .....</b>	<b>3</b>
<b>2. Типове потребители .....</b>	<b>3</b>
<b>3. Основни потребителски случаи (use case-и) .....</b>	<b>4</b>
<b>4. Основни изгледи (main views) .....</b>	<b>5</b>
<b>5. Server Resource API .....</b>	<b>6</b>
<b>6. Използвани технологии.....</b>	<b>7</b>
6.1. Клиентска част .....	7
6.2. Сървърна част .....	7
<b>7. Нефункционални характеристики .....</b>	<b>8</b>
7.1. Сигурност.....	8
7.2. Scalability.....	8
<b>8. Дизайн .....</b>	<b>9</b>
8.1. Дизайн на регистрацията .....	9
8.2. Дизайн на login .....	10
8.3. Дизайн на страницата за показване на постове .....	11
8.4. Дизайн на страницата за показване на потребителите .....	12
8.5. Дизайн на страницата за показване на коментарите .....	13
8.6. Дизайн на страницата за показване на постове по търсен hashtag .....	14
8.7. Дизайн на страницата за разглеждане на собствените постове.....	14

## 1. Идея на проекта

В днешно време едни от най-известните сайтове, са социалните мрежи (Facebook, Twitter, Instagram, Google+ и др.) . Всички тези сайтове след време започват да се разширяват в много и различни насоки и по този начин първоначалната идея залегнала в тях остава на заден план.

Проектът Hirundo има за цел точно това, да предостави на потребителите социална мрежа, в която комуникацията между тях самите ще бъде в центъра на вниманието. Социална мрежа, която да заеме освободилата се ниша, и да даде алтернатива на по-горе изброените гиганти.

## 2. Типове потребители

- Анонимни потребители

Това са потребителите, които нямат регистрация в сайта. Единствената функционалност, която им се предоставя е да се регистрират.

- Регистрирани потребители

Това са потребителите, които вече имат създадена регистрация в Hirundo. Преди да се впишат сайта, им се предоставя възможност за вписване като единствена функционалност.

- Вписани потребители

Това са същинските потребители на Hirundo. Те могат да следват или да престанат да следват други потребители. Виждат съобщенията на всички последвани потребители, да създават коментари към тях, да ги харесват и да премахват добавено харесване. Може да следи броя харесвания към някое съобщение. Също така този тип потребители имат възможност да създават свои съобщения, които да споделят със своите последователи. Постовите им могат да съдържат в себе си hashtag-ове. В случай, че вписан потребител иска види всички съобщения съдържащи конкретен hashtag (независимо дали съобщението е на последван потребител или не), тази възможност е налична в отделна страница.

### 3. Основни потребителски случаи (use case-и)

Потребителски случай	Кратко описание	Участващи потребители
Регистриране	Анонимен потребител, може да се регистрира в системата предоставяйки потребителско име, валиден email адрес и парола.	Анонимен потребител
Вписване	Регистриран потребител, може да се впише в системата предоставяйки своя email и парола.	Регистриран потребител
Следване на потребител	Вписан потребител, може да последва друг потребител, което му дава възможност да вижда съобщенията публикувани от втория.	Вписан потребител
Прекратяване следването на потребител	Вписан потребител, може да спре да следва друг потребител, като съответно спира да вижда публикуваните от него съобщения.	Вписан потребител
Публикуване на съобщение	Вписан потребител може да публикува съобщение. В съобщението си може да отбележи и hashtag-ове свързани с него.	Вписан потребител
Преглед на съобщения	Вписан потребител може да вижда публикуваните съобщения от потребители, които той следва.	Вписан потребител
Преглед на собствените	Всеки потребител може да разглежда своите публикувани съобщения.	Вписан потребител

съобщения		
Харесване на съобщение	Вписан потребител може да харесва съобщения.	Вписан потребител
Отказ от харесване на съобщение	След като вписан потребител е харесал дадено съобщение, той има възможност да се откаже от харесването.	Вписан потребител
Преглед на съобщения по hashtag.	Вписан потребител може да филтрира всички съобщения съдържащи конкретен hashtag (независимо дали следва авторът на даденото съобщение).	Вписан потребител
Добавяне на коментар към съобщение	Всеки вписан потребител има възможност за добавяне на коментар към дадено съобщение, на някой от последваните от него потребители.	Вписан потребител

## 4. Основни изгледи (main views)

Име на изгледа	Кратко описание	URI
Регистриране	Предоставя форма за регистрация, давайки възможност на даден потребител да се регистрира.	/register
Вписване	Предоставя форма за вписване в системата.	/login
Главен списък със съобщения (Feed)	Визуализира всички съобщения на потребители следвания от текущо вписания потребител в системата. Също така предоставя възможност за публикуване на ново съобщение	/feed
Списък със съобщения филтрирани по hashtag (Hashtag feed)	Визуализира всички съобщения съдържащи конкретен hashtag.	/hashtag_feed
Списък със собствени съобщения	Визуализира всички съобщения публикувани от текущо вписания потребител	/my_posts

## 5. Server Resource API

В таблицата отдолу са описани само използваните endpoint-и. Сървърът поддържа още.

Име на ресурса	Кратко описание	URI
Auth (login)	<b>POST:</b> Дава възможност потребител да се впише в системата предоставяйки email адрес и парола	/auth/login
Users	<b>GET:</b> Връща данни за всички регистрирани потребители в системата <b>POST:</b> Позволява създаването на нов потребител в системата	/users
User	<b>GET:</b> Връща данните за потребителя с конкретно userId	/users/{userId}
	<b>POST:</b> Позволява харесване на съобщение с дадено postId	/users/like
	<b>POST:</b> Позволява на потребител да спре да харесва съобщение с дадено postId	/users/dislike
	<b>POST:</b> Позволява на вписания потребител да последва потребител с дадено userId	/users/follow
	<b>POST:</b> Позволява на вписания потребител да спре да следва потребител с дадено userId	/users/unfollow
Posts	<b>GET:</b> Връща данните за всички съобщения, създадени от потребители следвани от текущо вписания потребител. <b>GET</b> (с query hashtag): Връща данните за всички съобщения съдържащи конкретния hashtag. <b>POST:</b> Създава ново съобщение.	/posts
	<b>GET:</b> Връща данните за съобщенията на текущо вписания потребител.	/posts/own
Comments	<b>GET</b> (с query postId): Връща данните за всички коментари към съобщение с посоченото postId. <b>POST:</b> Позволява създаването на ново съобщение.	/comments
Comment	<b>GET:</b> Връща данните за коментар с конкретно commentId	/comments/{commentId}

## 6. Използвани технологии

### 6.1. Клиентска част

- React – изграждане на структурата на UI-а на системата;
- React-router – навигиране между различните изгледи на системата;
- Redux – управление на данните по време на изпълнението на клиентската част на системата;
- Material UI – изграждане на красив и модерен UI на системата;
- Bluebird – библиотека за promise-и;
- Lodash – библиотека предоставяща множество функции за работа с обекти, списъци и стрингове;
- Superagent – библиотека за изпращане на заявки към сървъра;

### 6.2. Сървърна част

- Express – библиотека улесняваща изграждането на NodeJS приложения;
- Passport – библиотека даваща възможност за лесно изграждане на аутентикация в NodeJS приложения;
- jsData – библиотека за управление на данните в сървърната част на системата. Лесно изграждане на моделен слой;
- jsData-schema – библиотека за валидация чрез схеми на данни пристигащи към сървъра;
- Lodash – библиотека предоставяща множество функции за работа с обекти, списъци и стрингове;

## 7. Нефункционални характеристики

### 7.1. Сигурност

- Валидация на данните

При създаването на всеки ресурс се създава схема, спрямо която биват валидирани всички пристигащи данни от страна на клиентската част на системата;

- Наличие на проекции (projections)

За всеки ресурс могат да се дефинират, множество проекции. При прилагането на дадена проекция, данните, които се изпращат към клиента се филтрират спрямо отбелязаната проекция.

### 7.2. Scalability

- Лесно конструиране и изграждане на заявки към сървъра

Наличието на service-а Connection, осигурява лесен начин за изпращане на заявки към сървъра. API-то му предоставя функция send, чрез която се извършва самото изпращане. Функцията приема като аргументи endpoint-а, данните които да се изпратят и тип на заявката. Ако заявката не е от тип POST или PUT данните се добавят като query-параметри на заявката.

- Лесно добавяне на нови модели (ресурси) на сървъра

Основните endpoint-и като Collection Level Resource и Individual Level Resource са отделени в самостоятелна функция, която може да закачи съответните функции, които да слушат за дадените заявки от страна на клиентската част. При нужда функцията предоставя и възможност за конфигуриране (модифициране) поведението на съответните listener-и.

- Лесно превключване от един persistence layer към друг

Благодарение на библиотеката JSData използвана при реализацията на



сървъра на системата, в случай на нужда сървърът съвсем лесно може да бъде превключен да работи с друг persistence слой.

## 8. Дизайн

Дизайнът на сайта е реализиран спрямо най-съвременните стандарти в уеб приложенията. Използван е стандарта Material Design наложен от Google през последните няколко години.

Този тип дизайн се характеризира с изчистен външен вид, в който се цели да се открояват главно акцентите в съдържанието на дадения сайт. Същевременно съдържанието се наслагва в координатната ос “z”, така че да се реализира многослойност на съдържанието. Основна част в този аспект на дизайна е наслагването на пластове хартия един върху друг.

### 8.1. Дизайн на регистрацията

Дизайнът за изгледа за регистрация използва стандартни TextField компоненти от Material-UI

The screenshot shows a web browser window with the address bar displaying 'localhost:9988/register'. The page has a blue header with the text 'Hirundo' and a hamburger menu icon. The main content area contains a registration form with the following fields: 'Email' (filled with 'maria@mail.com'), 'Username' (filled with 'maria'), 'Password' (filled with five dots), and 'Password repeat' (filled with five dots). Below the fields is a blue 'REGISTER' button. The browser's taskbar at the bottom shows various application icons.

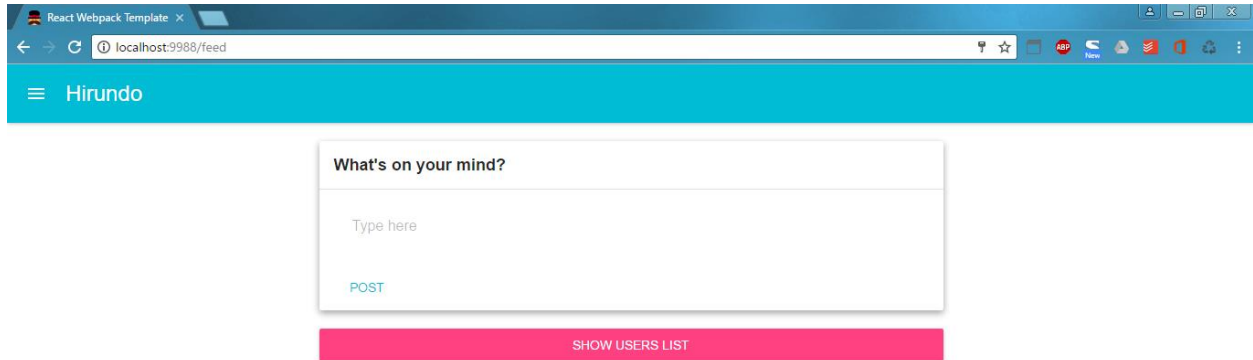
## 8.2. Дизайн на login

Този изглед използва същите ресурси, като по-горния, но са нужни по-малко полета:

The screenshot shows a web browser window with the address bar displaying 'localhost:9988/login'. The page has a blue header with the text 'Hirundo' and a hamburger menu icon. The main content area contains a login form with the following fields: 'Username' (filled with 'maria') and 'Password' (filled with five dots). Below the fields is a blue 'LOGIN' button. The browser's taskbar at the bottom shows various application icons.

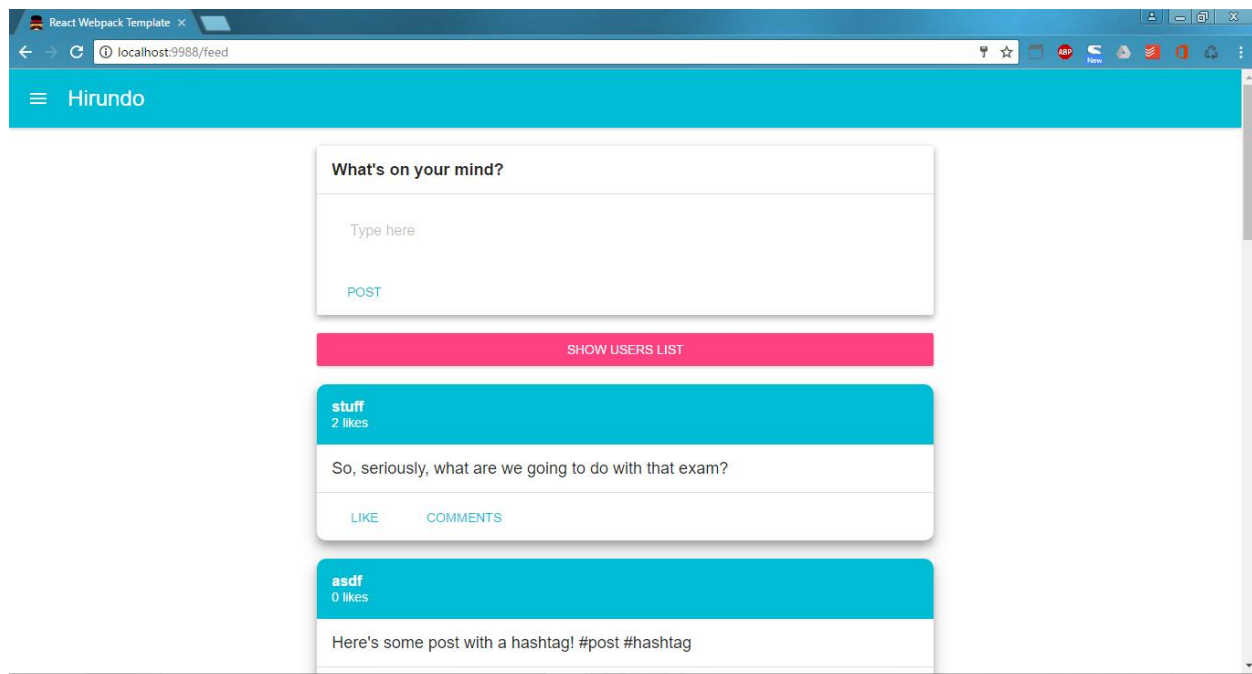
### 8.3. Дизайн на страницата за показване на постове

Новосъздаден потребител би имал следния изглед:



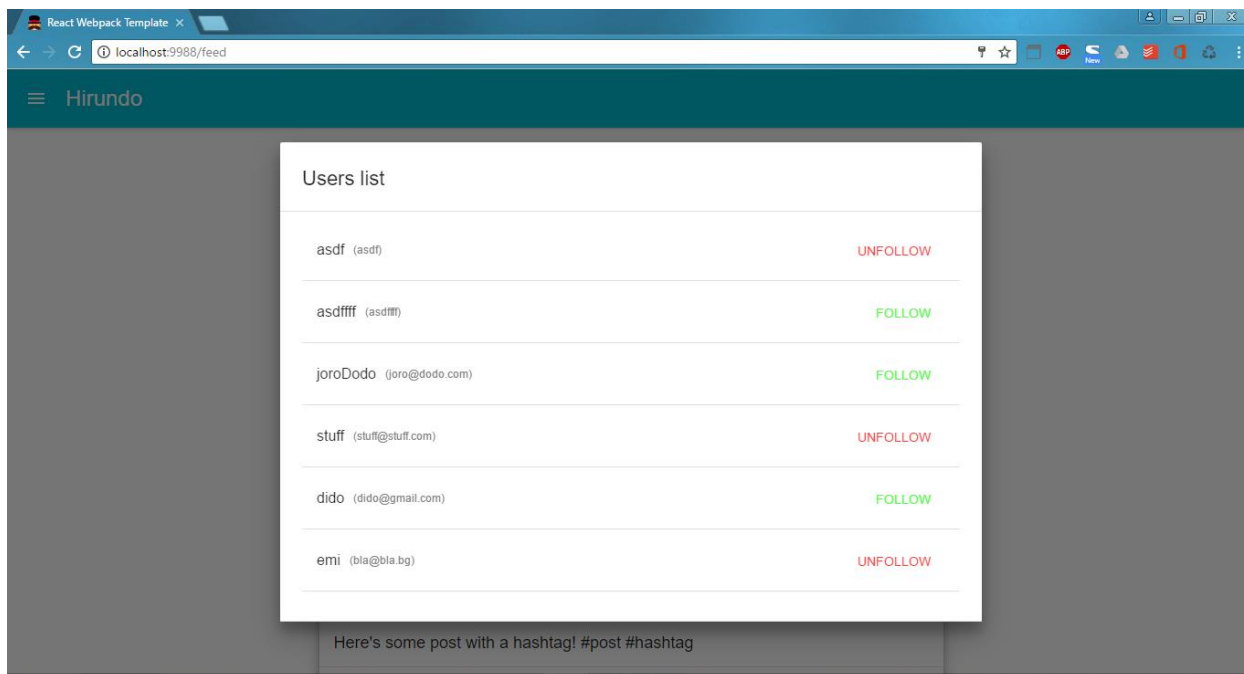
---

При последване на други потребители той ще има следния изглед:



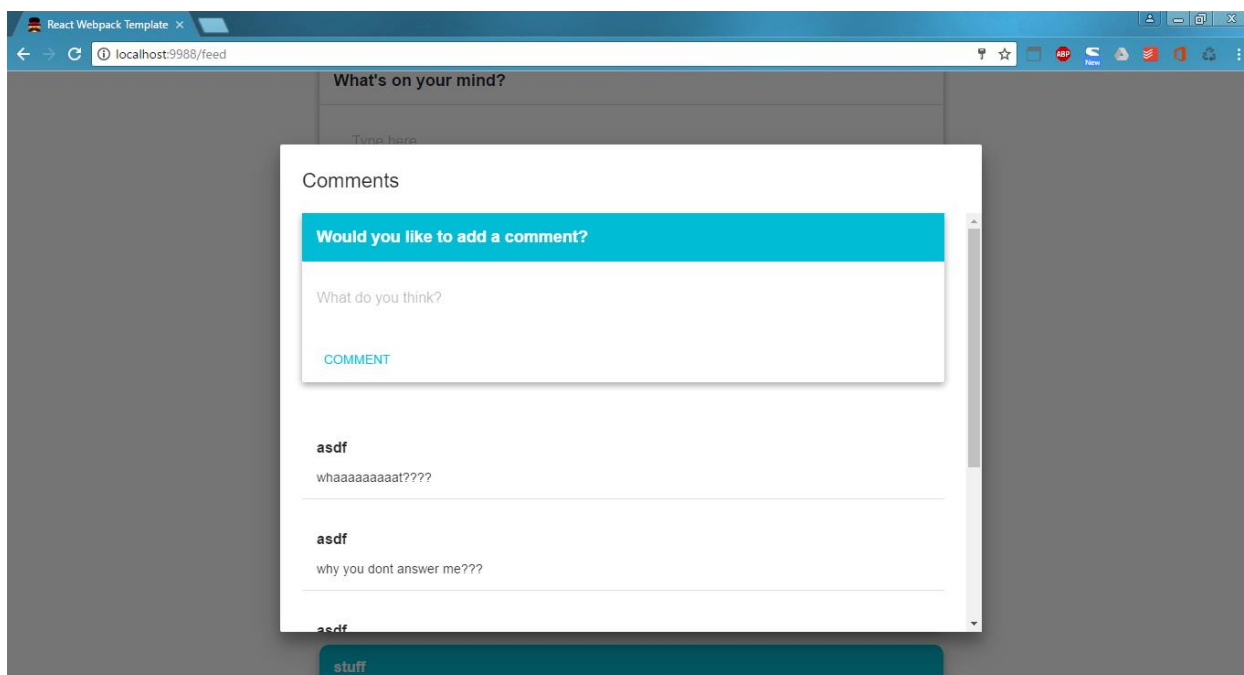
#### 8.4. Дизайн на страницата за показване на потребителите

Създава се диалог след натискане на “SHOW USERS LIST”, в който може да се последват потребители. Това тяхно последване се отразява веднага на показваните в момента постове.

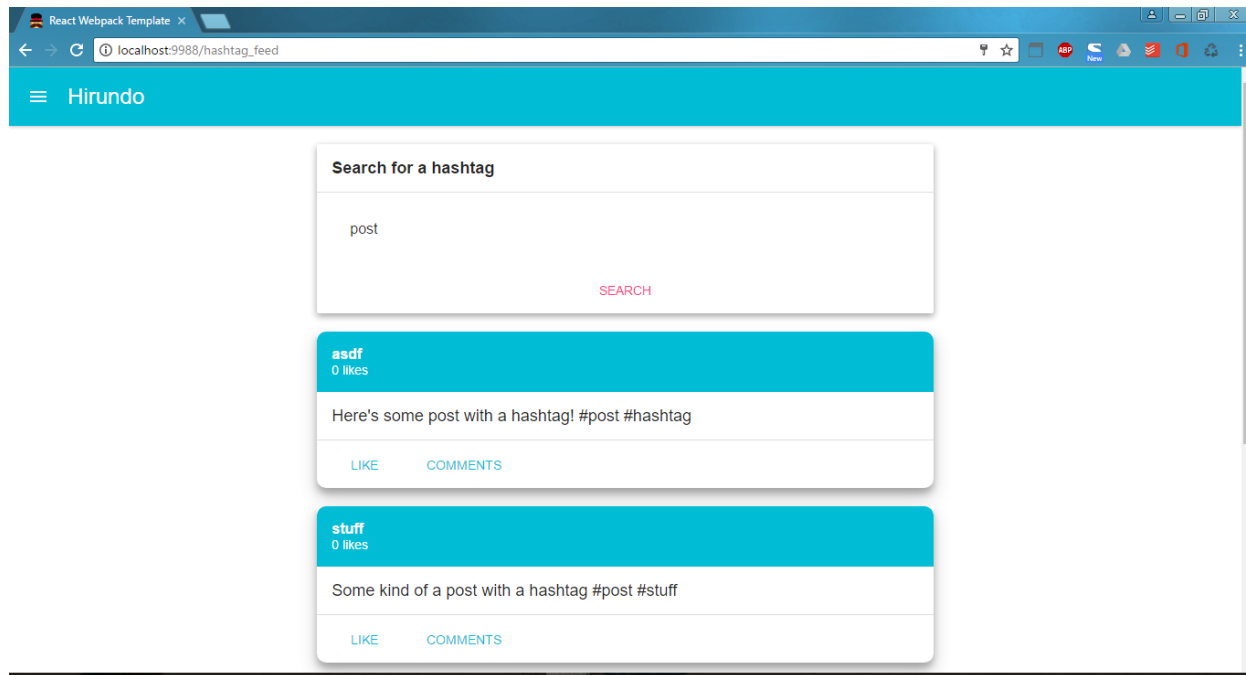


## 8.5. Дизайн на страницата за показване на коментарите

Също се създава диалог, в който се показва компонент за създаване на коментари към пост, а под нея и вече съществуващите коментари.



## 8.6. Дизайн на страницата за показване на постове по търсен hashtag



## 8.7. Дизайн на страницата за разглеждане на собствените постове

