|  |  |
| --- | --- |
|  | untitled |

**AKYC Social Media Website**

Bachelor’s Thesis

Yaren Çoşkun 217SE2003I

Ali Haydar Konuk 216CS2411D

Supervised by

Günet Eroğlu

January 2020

Abstract

Social media has started to become one of the main determinants of human relations in recent years. It occupied an important place in people's lives as a virtual environment used for daily conversations, sharing information, discussions, following local or global daily developments, and even to clear the head at leisure time. As a result, people were empowered to have access to any opinion on any topic. However, this authority brought a problem with it. Although people follow and share in this common network based on their own tastes, hobbies and fields of activity, these special areas of interest have generally disappeared within the large common network. The increase of social media users has made it difficult for people to follow current developments and ideas about their own field of interest. Eventually, social media applications that share in a specific area, isolated for specific topics, became a need.

Although there are many websites on the internet in terms of gastronomy and culinary arts, which are the special areas of our project, none of them aim to create a private network on this subject. Our project basically does not have more than the general functions of any social media application. The interesting point of our website for the external user is that it addresses to a limited number of people.

The fact that social media is an intermediary for the flow of data and ideas on a global scale will make it necessary to create isolated common networks both within our topic and other areas of interest in the future.

**ACKNOWLEDGEMENTS**

We would like to acknowledge Asst. Prof. Günet Eroğlu as supervisor that helped us with their guidance throughout the project.

Table of Contents

[1. INTRODUCTION 6](#_Toc62406276)

[2. LITERATURE REVIEW 2](#_Toc62406277)

[2.1. The History of Social Media Platforms on Network 2](#_Toc62406278)

[2.2. Shortcomings in Social Media 3](#_Toc62406279)

[2.3. About the Project 3](#_Toc62406280)

[2.4. Current System 3](#_Toc62406281)

[2.5. Fuzzy Search 4](#_Toc62406282)

[2.6. Bcrypt Algorithm 4](#_Toc62406283)

[3. PROPOSED SYSTEM 5](#_Toc62406284)

[3.1. Introduction 5](#_Toc62406285)

[3.2. Functional Requirements 5](#_Toc62406286)

[3.3. Nonfunctional Requirements 5](#_Toc62406287)

[3.4. Current Software Architecture 7](#_Toc62406288)

[3.5. System Decomposition 8](#_Toc62406289)

[3.6. Access Control and Security 9](#_Toc62406290)

[4. Implementation, tests, experiments 10](#_Toc62406291)

[4.1. Implementation 10](#_Toc62406292)

[4.2. Tests 10](#_Toc62406293)

[4.3. Experiments 20](#_Toc62406294)

[5. conclusions and future work 22](#_Toc62406295)

[6. References 23](#_Toc62406296)

LIST OF FIGURES

[Figure 1 System Decomposition Map 8](#_Toc62415132)

[Figure 2 Register Page 11](#_Toc62415133)

[Figure 3 Login Page 11](#_Toc62415134)

[Figure 4 Main Page 12](#_Toc62415135)

[Figure 5 Create a Post Pop Up 12](#_Toc62415136)

[Figure 6 Like Post (1) 13](#_Toc62415137)

[Figure 7 Like Post (2) 13](#_Toc62415138)

[Figure 8 Add Comment (1) 13](#_Toc62415139)

[Figure 9 Add Comment(2) 14](#_Toc62415140)

[Figure 10 Search Page 14](#_Toc62415141)

[Figure 11 Follow (1) 15](#_Toc62415142)

[Figure 12 Follow (2) 15](#_Toc62415143)

[Figure 13 Settings - User Information Page 16](#_Toc62415144)

[Figure 14 Delete Post (User) 16](#_Toc62415145)

[Figure 15 Delete Post (Admin) 17](#_Toc62415146)

[Figure 16 Delete Comment (User) 17](#_Toc62415147)

[Figure 17 Delete Comment (Admin) 18](#_Toc62415148)

[Figure 18 Settings - Verification Request Page 18](#_Toc62415149)

[Figure 19 Admin - Verification Requests Page 19](#_Toc62415150)

[Figure 20 Forgot Password Page 19](#_Toc62415151)

[Figure 21 New Password Mail 20](#_Toc62415152)

# INTRODUCTION

Social media applications have taken a serious place in people's lives for more than 20 years, especially with the introduction of internet networks into people's daily lives in recent years. The Internet has caused a radical change in people's habits of being aware of daily news, social communication, and information research. Internet has prevailed over print publication and has gradually become an alternative for literary works. On the other hand, even though the internet is thought to occupy the expansive majority of people's lives, there are still areas that the internet cannot populate today.

To give an example of one of these, in terms of gastronomy and culinary arts, which are our field of work, there is no participant, information sharing and discussion platform on the internet by external users. The websites that exist in this area are those that only share encyclopedic information or give recipes, usually one-sided, not interactive with the visitors.

Social media application administrators generally do not prefer to limit the posts on their websites to a specific subject in order to keep the user spectrum at the highest possible level. As a result, people who have the habit of using social media applications frequently have to spend too much concentration and effort when they look for posts on a specific area and try to keep themselves updated in this area.

However, there are also social media applications that specifically address a specific issue and restrict their users to share within the framework of this topic. Among them, the most popular one allows sharing in terms of general skeleton, but in terms of the general concept of the website and active users within the website, it encourages the end user to share the general concept of the website that academia and business life.

Similar to the social media application we mentioned above, we decided to design a social media website with a specific concept of gastronomy and culinary arts. Like any social media application, the website will be a platform that will allow you to get an account, share post, like shared post, comment on posts, follow and similar features. Considering both the role of social media in people's lives and this field is untouched in terms of social media, we think that our project will meet this need.

# LITERATURE REVIEW

## The History of Social Media Platforms on Network

Although the emergence of social media is much older, as we understand it today, social media applications emerged with Facebook, developed by Mark Zuckerberg in the early 2000s. With social media applications such as Facebook, people have gained identity for the first time through an account on internet networks. Internet networks began to be used more interactively with this process, and today these platforms have become a simple model of daily life on social networks.

Twitter was founded by Jack Dorsey, Noah Glass, Biz Stone, and Evan Williams in 2006 as the first major alternative to Facebook. While Facebook is a social media platform where people communicate with their friends, family and other immediate surroundings, Twitter provides an environment where members can establish a more comfortable interactive relationship and interact with each other regardless of their communication with each other. While Facebook offers its members a more local network that they can limit themselves, Twitter in turn offers its users a more global network. This difference has led Twitter to grow as an important alternative to Facebook. (What is the Difference Between Twitter and Facebook?)

Instagram was founded in 2010 by Kevin Systrom and Mike Krieger. Instagram is an app for sharing photos and short videos, and made its debut as a free photo editing and sharing app. In an era where visuality is important and remarkable, Instagram is one of the most visual-oriented social networks. Due to the fact that the posts on Twitter have a faster flow and the boundaries between the users on Twitter are more permeable, there was a need for a social media platform where local areas are more prominent. Instagram has taken its place among the social media applications as a third alternative because it corresponds to this need and is a visual sharing based social media application. (Koçoğlu)

## Shortcomings in Social Media

The applications we have mentioned so far are social media platforms that have avoided focusing on a specific area and try to keep the user scale at the highest level they can keep. Even if the users create communities specifically for their interests within the application or begin to follow users who share these interests, it required a special effort to ensure the continuity of this. Although the effort required for this continuity has decreased a little more with the groups that can be created on Facebook, this grouping feature has not been in demand by social media users. Apart from all these, websites that share culinary arts are generally platforms that either share encyclopedic information or recipes that can be useful in daily life, do not have or have very limited interactions with or between visitors. As an exception, LinkedIn has become an example of an active social media application that appeals to a specific segment in a specific field by restricting its users to post only on academia and business life.

## About the Project

In fact, our main motivation during both the project decision-making process and the further process of the project is that the idea we have created does not have an active response today. As we have stated on the upper side, we believe that our application will correspond to some needs arising from the contradictions of social media applications. Since our application has a special concept, it will appeal to a more limited audience, unlike other social media applications. As a positive result of this, our social media application will be identified with this concept after a certain period. And finally, our website will be the first address for people who want to learn or share information about it.

## Current System

Most of the social media website has no specific concept. In this type websites, Users creates their own content such as humor, daily news, personal using and go on. Some social media websites have specific concept like LinkedIn. However, there is no social media website about gastronomy, culinary art, recipe, or food history.

In this point, we noticed that there is not a free platform which anyone can share post with their own ideas or their knowledge about gastronomy and culinary arts. (Konuk ve Çoşkun, Requirements Analysis Document)

## Fuzzy Search

Fuzzy search is a search algorithm that includes results that do not exactly correspond to the searched words in the total results. In other words, using fuzzy word search, it also returns results that may be related to the searched word alongside results which contain searched words. (Rouse)

So, we used fuzzy search in our project for searching users. When end-user search a word if username of any user contains this word, website displays this user even though searched word and username are not exactly same.

## Bcrypt Algorithm

Bcrypt is a hashing method to encrypt password. For OpenBSD and other systems which are like some Linux distributions such as SUSE Linux, the bcrypt function is the default password hash algorithm. We use Bcrypt in our project to protect user's password. Through hashing, even someone accesses to database he/she cannot learn any user's password. (Bcrypt)

# PROPOSED SYSTEM

## Introduction

The system provides a social media platform that users can register to website. After, they can share post. Also, they can comment to post, and they can like to post. They can delete their posts or comments. They can update their own profile information. They can search other users. They can interact with other users with chat. Also, admin can delete posts or comments. Also, admin can verify users' profile.

In our project, we use NodeJS framework to build website by using JavaScript programming language. Also, we use ExpressJS web framework which based on NodeJS. Additionally, we use MongoDB for database, and we use MongoDB-Compass to manipulate database. (Konuk ve Çoşkun, Requirements Analysis Document)

## Functional Requirements

1. Visitor can just access to registration and login panel. He/she can register to website.
2. User can login to system. He/she can share post about any information of culinary art, recipe, regional foods, or something else after the login. Additionally, user can optionally add photo to their own post. Likewise, user can delete their own post. User can like any posts. Also, he/she can comment to any post. Moreover, user can delete their own comment. Besides all these, user can interact with other users by using Direct Message. User can update their own personal information. User can search other users.
3. Admin can delete users' posts and also admin can delete users' comments too if he/she thinks that this is inappropriate or irrelevant. In addition to these, admin can verify users' account. (Konuk ve Çoşkun, Requirements Analysis Document)

## Nonfunctional Requirements

**3.3.1 Usability**

• Anyone with access to the internet can easily understand and use the functions of the website through understandable interface.

• The user interface is essentially a typical social media website interface.

• Frequently Asked Questions (FAQ) is included in the website.

**3.3.2 Reliability**

• User information cannot be shared without sharing permission from user.

• The maintaining is performed without restarting of the system when a failure event is happened.

• Data are periodically backing up. Therefore, a important data loose will be not possible.

• User is redirected to an error page which contains possible error reasons when an exception is happened.

• User can change their password through using email if user forgot their password.

• All passwords are encrypted when user fills and sends the registration form.

**3.3.3 Performance**

• The website pages change responsively for each device such as smartphone, tablet or computer.

• The transmission of the message to another user should be in an acceptable time since messaging between users is an action that needs to be performed quickly.

• There can be no restrictions on the online users’ number since our project is a social media platform.

• 2 seconds for registration and login and 5 seconds for searching are acceptable worst latency time.

**3.3.4 Supportability**

• .png, .jpg and .jpeg extensions are supported for profile picture and picture with post.

• Developers maintain the system with monthly period for encountered exceptions.

• The system is a website. The system doesn’t have any android or IOS application.

**3.3.5 Implementation**

• Our project can be used in only device which has internet access.

• The source code of the project is written in JavaScript by using NodeJS Framework.

**3.3.6 Interface**

• Any other existing systems don’t interact with our project.

• Data importing is happened with registration and user action inside the website.

**3.3.7 Packaging**

• Developers of the website install the system.

• Node package managers in NodeJS designates number of installations.

**3.3.8 Legal**

• Our project will be licensed with public domain.

• Website HTML template is purchased. (Konuk ve Çoşkun, Requirements Analysis Document)

## Current Software Architecture

In the project, we used Single Thread Event Loop Model as an architecture. General structure of the architecture is that. All requests from the client side are handled with a single thread. This thread is named as 'Event Queue'. The web server has an internal component also known as 'Event Loop'. The function of this loop is that receiving user requests and processing these requests. Event Loop checks whether requests from the client side in Event Queue. If there are any request, it takes any request from Event Queue. If not, Event Loop continuously waits for a user request. In case of user request, Event Loop selects an available threat from Thread Pool. Then, Event Loop gives incoming user request to selected available thread and Event Loop assigns the thread with this request. Then, thread takes this request and process this. After that, it executes to block IO operations if necessary. It prepares a user response to return client side. It sends to Event Loop. Event Loop returns the responses to client side, respectively. (Konuk ve Çoşkun, System Design Document)

## System Decomposition

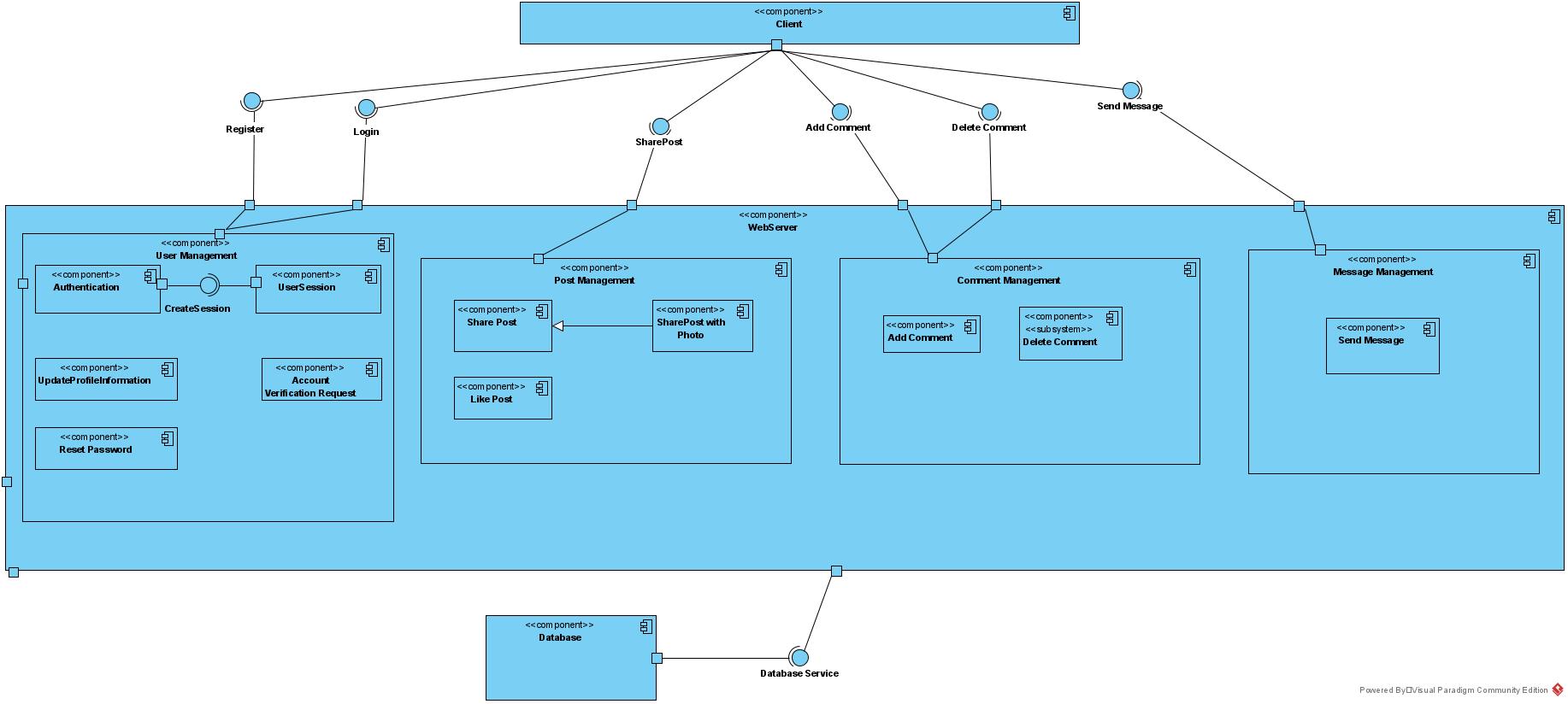
****

Figure System Decomposition Map

In our project, we used 'Single Thread Event Loop Model' Architecture. The project is essentially broken into three layers. These are Client, Server and Database. In Client layer, there is interface that user can trigger functions of the website. Second layer is Server. In this layer, server has essentially four different components such that UserManagement, PostManagement, CommentManagement and MessageManagement. First component UserManagement is responsible for authentication, updating of user profile, verification request for own account and resetting own password. Second component PostManagement is responsible for sharing post, sharing post with photo, and liking post. Then, CommentManagement is responsible for adding and deleting comment. Lastly, MessageManagement is responsible for chatting. The last layer is Database. It stores incoming data from Server. (Konuk ve Çoşkun, System Design Document)

## Access Control and Security

Access boundary between actors and objects with related functions is given in the table below. (Konuk ve Çoşkun, System Design Document)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| OBJECTS  ACTORS | Post | Comment | Request | Account | Message |
| User | addPost()  deletePost()  likePost() | addComment()  deleteComment() | verificationRequest() | updateProfileInformation()  resetPassword() | sendMessage() |
| Admin | deletePost() | deleteComment() | accountVerification() |  |  |
| Visitor |  |  |  | signIn()  signUp() |  |

# Implementation, tests, experiments

## Implementation

Our project mainly consists of 3 parts in terms of architecture. These; Client, Server and Database. We used MongoDB as the database. One of the main reasons for this is that the schemas created in MongoDB are more flexible, more dynamic and editable. When using MongoDB, there are no schema structures that the data must comply with. They can be shaped according to the needs. Another important feature is that it is compatible with NodeJS. We used ExpressJS, a NodeJS based web application framework on the server side. We run our server on ExpressJS. We established the connection with MongoDB here and created our models with MongooseJS. With the help of these models, we created the schemas to be kept on MongoDB and edited these schemes in line with the needs of the project. We configured our APIs over ExpressJS routers and performed both our routing and CRUD operations through these APIs. On the client side, the third and last part of the project, we used a template engine called Handlebars. We handled the data from the API with Handlebars. Handlebars provided the opportunity to display this data using certain conditions and loops when necessary. Finally, we used JavaScript as the programming language in our project.

## Tests

For software testing, we manually tested the main functions that should be on the website.

* **Register**

When you click the 'register' button on the home page, the register page should be opened. It opened up as it should when clicked. When the username, name, surname, email, date of birth, city, job, password information were filled in completely and the terms and conditions were accepted, and finally when ‘register’ button at the bottom is pressed, the notification that “you have successfully registered” appeared. When we looked at our database table, we saw that the new user was successfully registered.

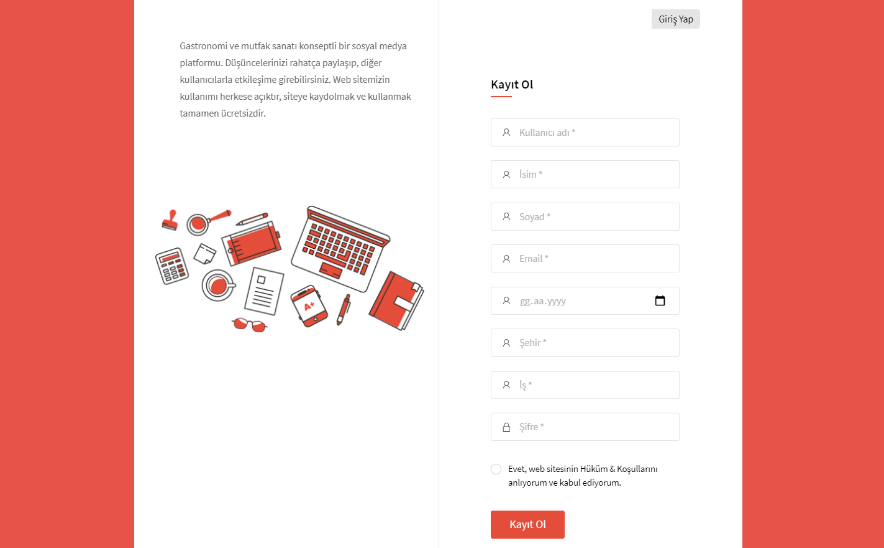


Figure Register Page

When we left a blank space on the information to be filled on the register page, we saw that the 'please fill in this field' warning appeared as it should be.

When we do not accept the terms and conditions on the register page, “To register, you must accept the website's Terms & Conditions!” notification appeared.

When we tried to register by typing a previously registered e-mail in the e-mail section on the register page, the warning "There was a problem while registering" notification appeared.

We have seen that register function works as it should. We did not encounter any errors during the tests.

* **Login**

When the username and password fields on the login page are filled completely and correctly, we expect the user to log in successfully.

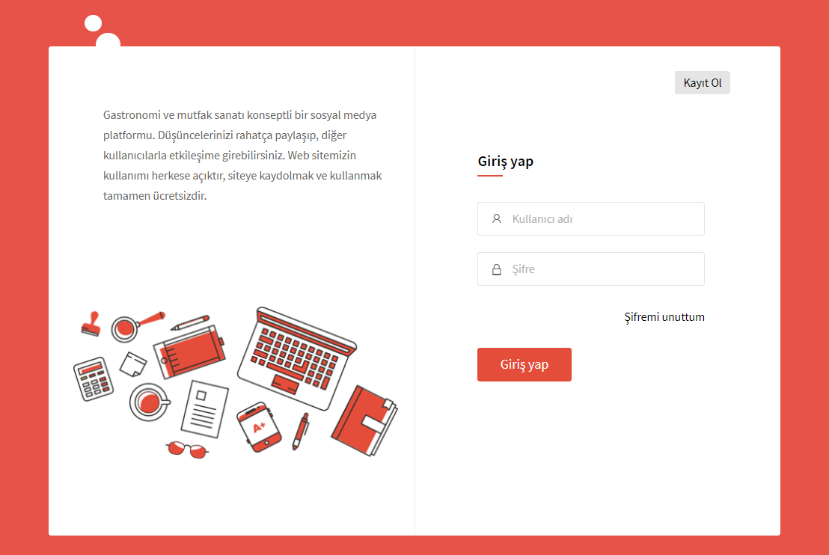


Figure Login Page

We have seen that when we enter the username and password of a registered user in the system, we can log in without any problems.

When we left a blank space on the information to be filled on the login page, we saw that the 'please fill in this field' warning appeared as it should be.

When we enter incorrect username and password information, we saw "Make sure you enter your username / password correctly!" warning appeared.

Consequently, we have seen that login function works as it should. We did not encounter any errors during the tests.

* **Share Post**

When the 'share' button on the Home Page is clicked, we expect a pop up with the title 'create post' to open. When the content part in this pop-up is filled and if desired, a photo is added to the post and the 'share' button is clicked, this new post is expected to be seen in the post flow on the homepage and in the user's profile.

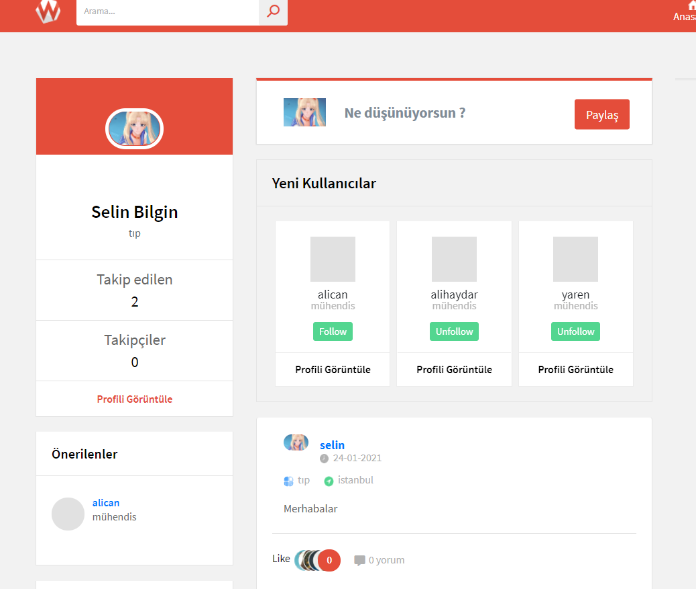


Figure Main Page

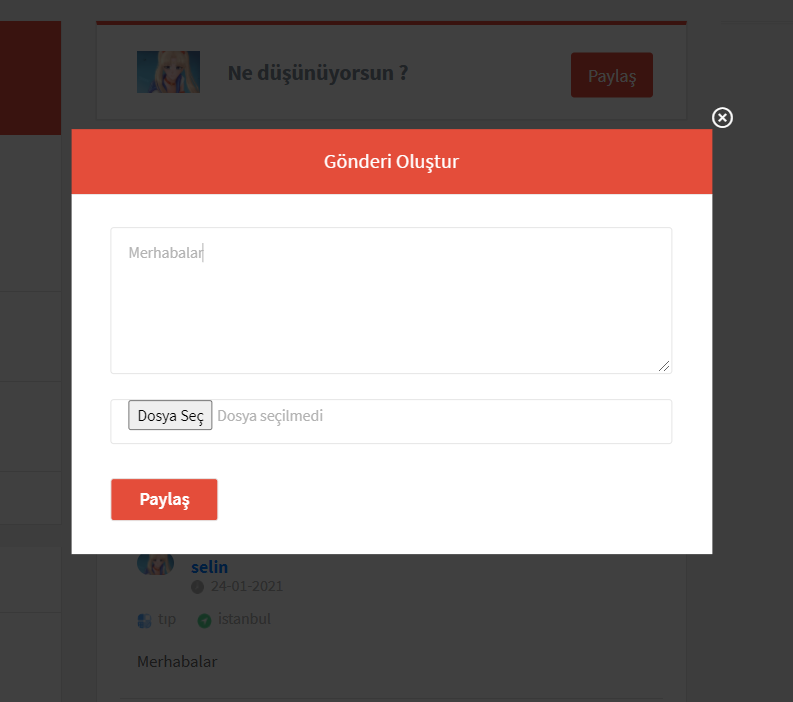


Figure Create a Post Pop Up

We clicked the 'share' button on the home page, filled in the content in the pop-up, then we pressed the share button in the pop-up. After that, we saw that our post was displayed smoothly on the homepage and on the user's profile. When we tried the process of creating a new post by adding a photo, we did not encounter any problems. The photo was successfully displayed in the post. When we looked at our database table, we saw that the new post was successfully stored.

* **Like Post**

When we click the like button, we expect the number of likes to increase by one, the like button turns into an unlike button and we expect our likes to remain the same when we refresh the page or move to another page.

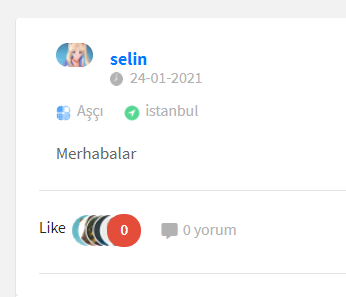
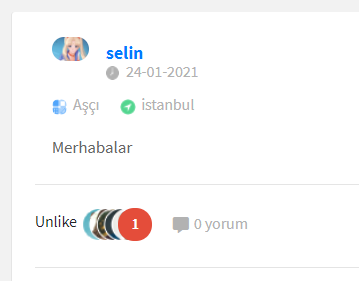
 

Figure Like Post (1) Figure Like Post (2)

When we clicked the like button on a shared post, we saw that the number of likes increased by one and the like button turned into an unlike button. When we refresh the page, we saw that our like is still remain. When we entered the page of the person who shared the post, we found that our like was also remain there.

* **Add Comment**

When a comment is made, we expect to see it under the relevant post.

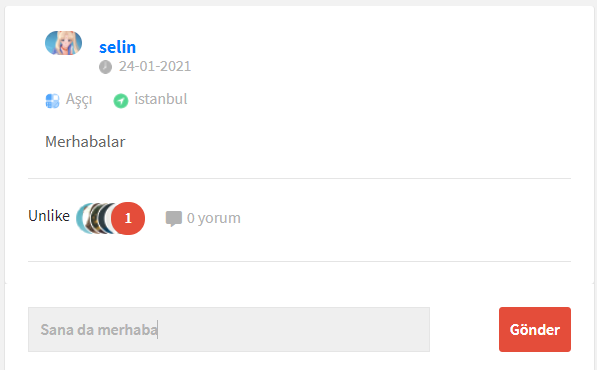


Figure Add Comment (1)

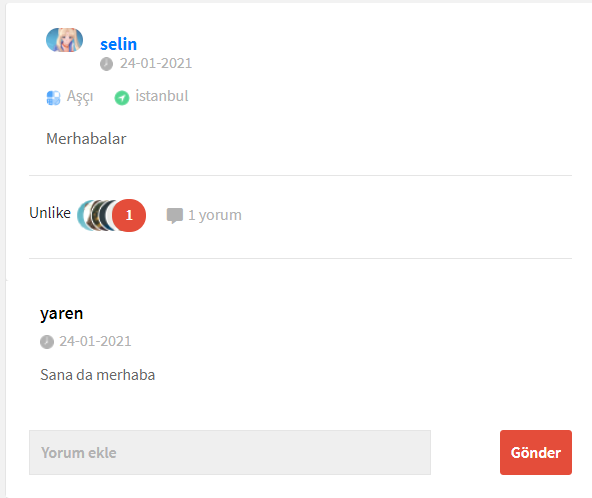


Figure Add Comment(2)

In the comment section under a post, we wrote a comment to the section that says 'add comment' and clicked the 'send' button. Afterwards, the comment we wrote under the post was successfully viewed. In addition, the posted comment started to appear under the My Comments section on the author's profile page. When we looked at our database table, we saw that the new comment was successfully stored.

* **Search**

When we use the search function, we expect results related to the words we enter as input.

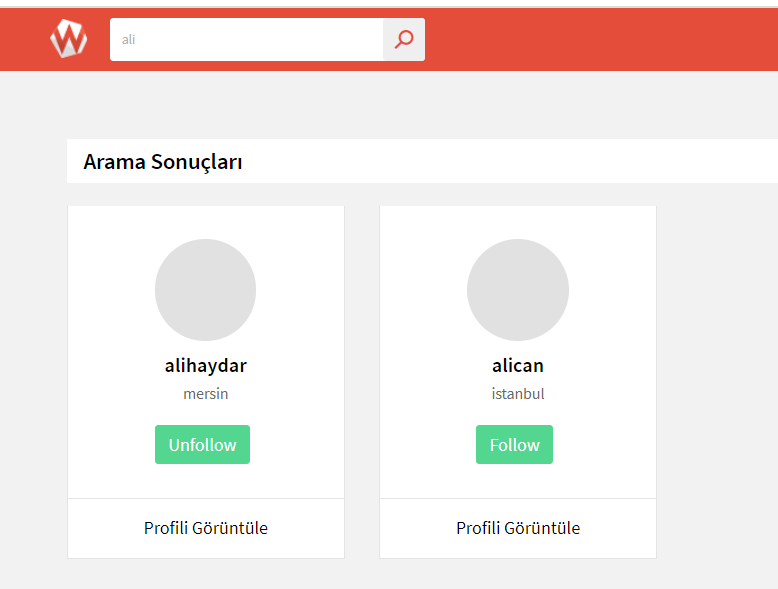


Figure Search Page

In exchange for the various words or letters we typed in the search bar, we saw that results containing those words or letters were successfully displayed. When we left the search bar blank without typing anything, we got a result showing all users in the system.

* **Follow User**

When we follow a user, we expect the 'follow' button belonging to the user to turn into an 'unfollow' button, the followed person's posts to be displayed in the post flow on the home page and the following event to be recorded in the database.

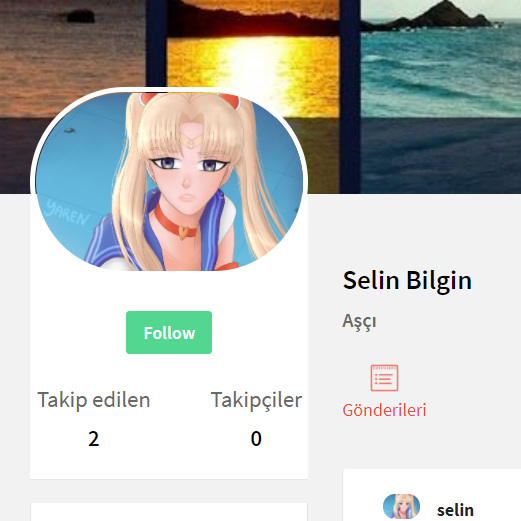


Figure Follow (1)

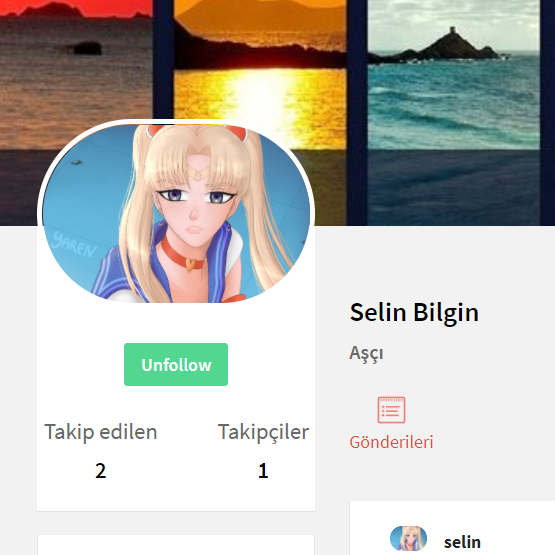


Figure Follow (2)

When we clicked ‘follow’ button and followed a user in the system, we saw that the 'follow' button belonging to the user turned into an 'unfollow' button, we saw that the posts of this person we are following are included in the post flow on the home page, we saw that the number of followers of the person followed increased by one and the number of followers of us increased by one and when we checked the database, we saw that this following event was successfully recorded.

* **Update Profile Information**

When a user wants to update their information, we expect that the information they have changed is registered in the database and that their information is changed and updated with new information wherever it appears on the website.

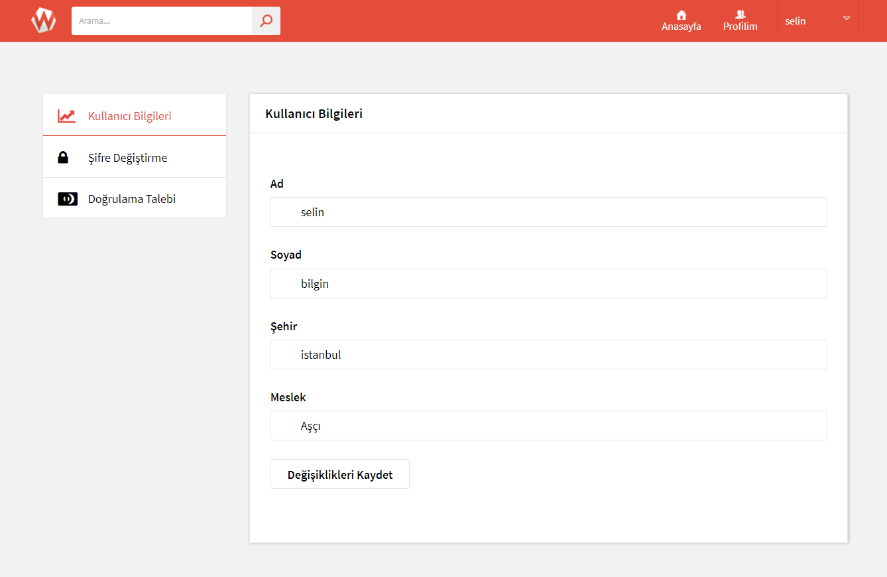


Figure Settings - User Information Page

In the User information tab on the Settings page, we changed the name, surname, city, and occupation information and pressed the 'save changes' button.

We saw that the information we changed was saved in the database, and that new information was updated on our profile page and then on our posts in the post flow.

* **Delete Post**

When a post is deleted, our expectation is to remove the post from both profile of the person who shared the post and the post flow on the home page.

Posts can be deleted by both the owner of the post and the admin.

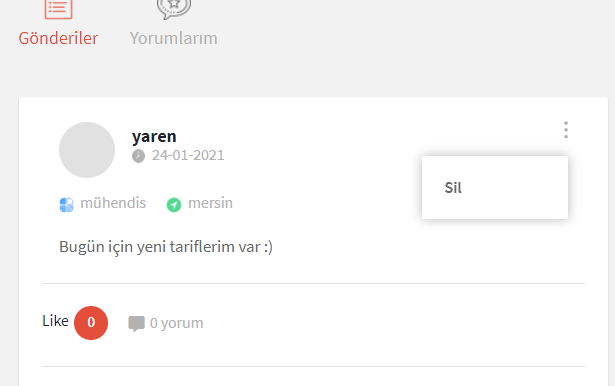


Figure Delete Post (User)

In order to be able to delete the post on the user's side, we logged into the user's profile. We clicked on the 'delete' option that appears when we click the '...' button on the upper right corner of the post we want to delete. It removed from both profile page and the post flow on the home page successfully.

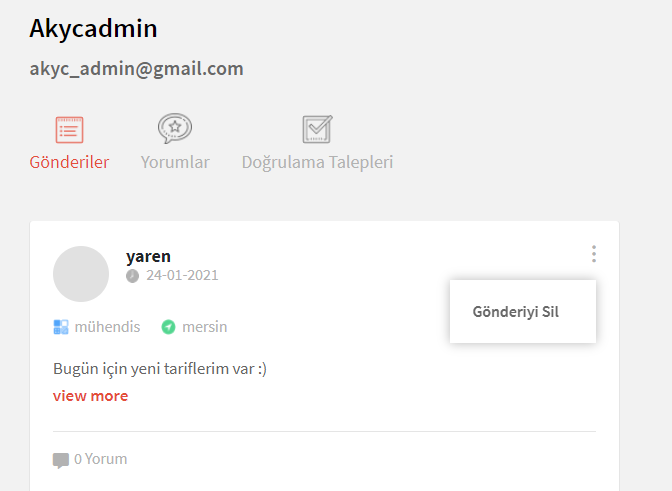


Figure Delete Post (Admin)

In order to be able to delete the post on the admin’s side, we logged into the admin account. On the admin page, in the posts tab, we clicked on the 'delete the post' option that appears when we click the '...' button on the upper right corner of the post we want to delete. It removed from both profile of the person who shared the post and the post flow on the users’ home page successfully.

* **Delete Comment**

When a comment is deleted, our expectation is to remove the comment from both the commenter's profile and the corresponding post.

Comments can be deleted by both the owner of the post and the admin.

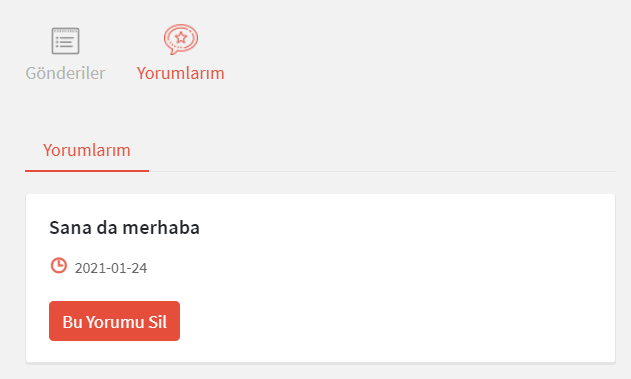


Figure Delete Comment (User)

In order to be able to delete the comment on the user's side, we logged into

the user's profile. We entered the My Comments tab and clicked the 'Delete this comment' button of the comment we want to delete. It removed from both the commenter's profile and the corresponding post.

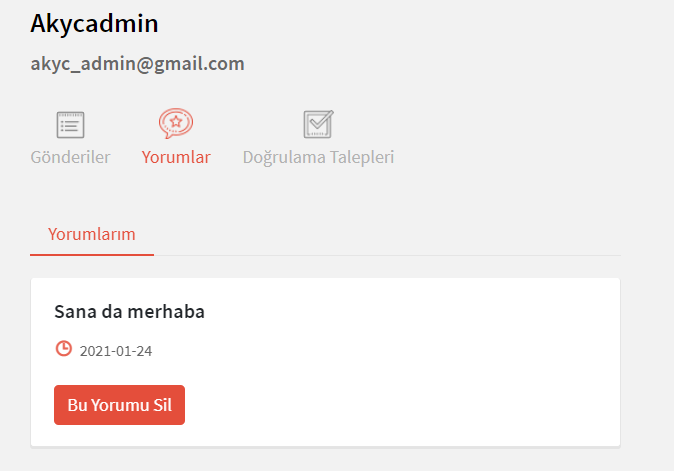


Figure Delete Comment (Admin)

In order to be able to delete a comment on the admin’s side, we logged into

the admin account. On the admin page, in the comments tab, we clicked the 'Delete this comment' button of the comment we want to delete. It removed from both the commenter's profile and the corresponding post.

* **Account Verification**

In account verification, we want it to be displayed on the admin page when the user sends a request. In the admin section, if this request is approved by the admin, a confirmation icon should appear on the account of the user who sent the request.

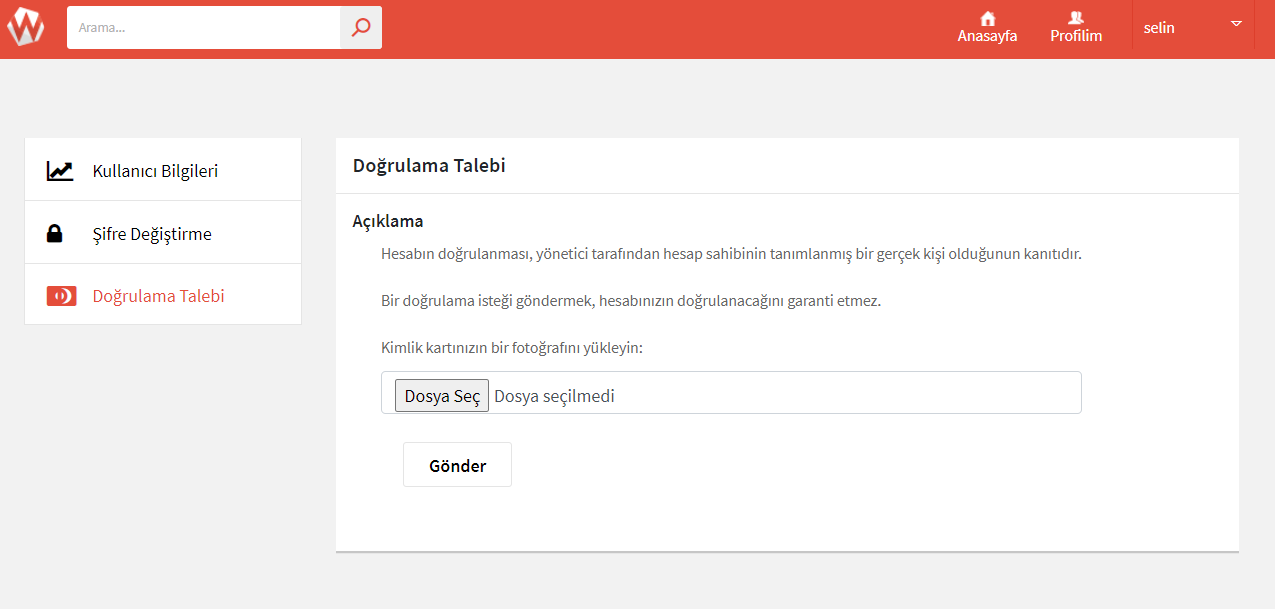


Figure Settings - Verification Request Page

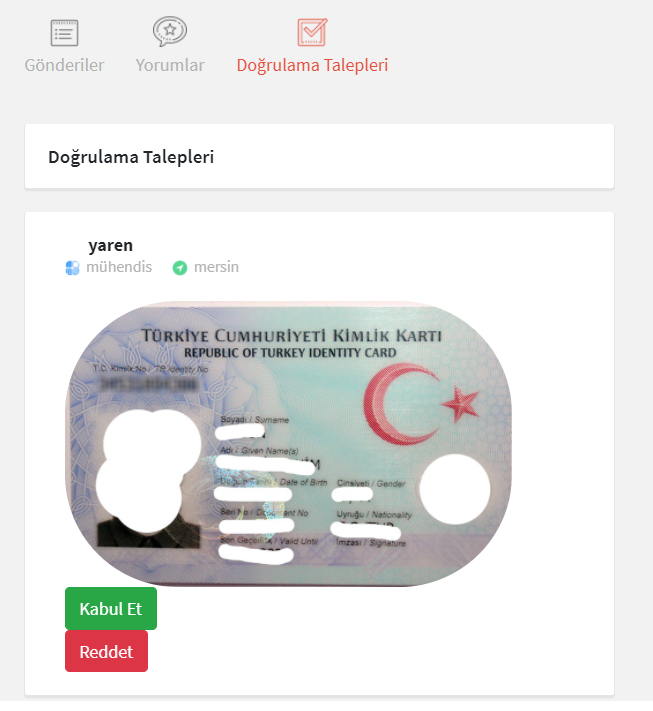


Figure Admin - Verification Requests Page

In the user account, we entered the verification request tab in Account settings. We uploaded the ID card photo and clicked the 'send' button. On the admin page, in the verification requests tab, we saw that the request we created appears. We clicked the 'accept' button next to the request. The request has been removed from the admin page. Confirmation icon appeared on the account of the user who sent the request.

* **Forgot Password**

When using the forgot password function, our expectation is to assign a new password to the user and inform the user, and this new password must be changed with the new password in the system.

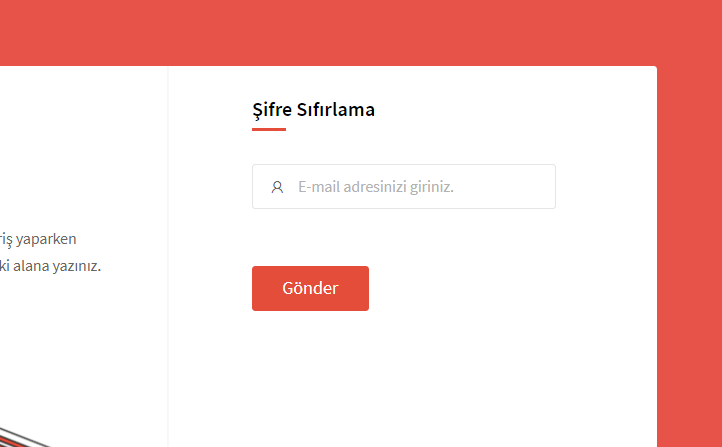


Figure Forgot Password Page

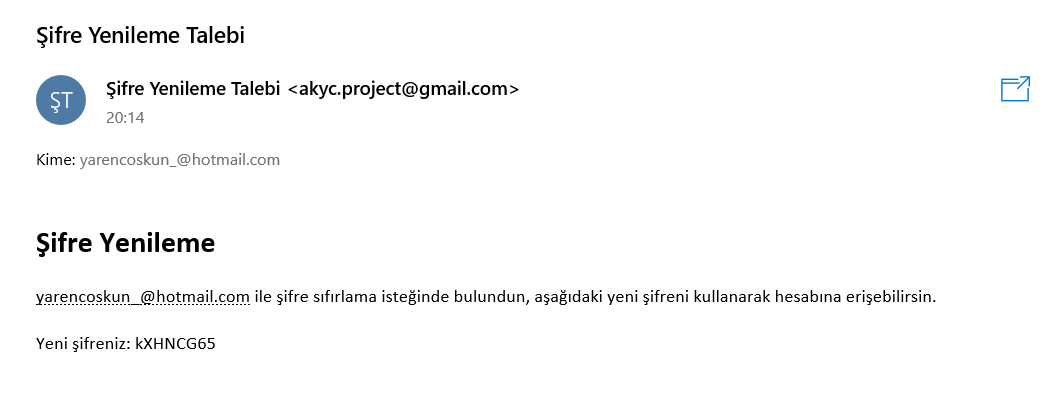


Figure New Password Mail

We clicked the 'forgot password' link on the home page. We entered an e-mail that is registered in the system that we want to reset the password. We clicked the Send button. An e-mail containing the new password was sent to the e-mail address we entered. We tried to log into the account with the new password and we were able to log in without any problems.

## Experiments

We've asked a person who is totally independent from the project to use the website. First, we noticed that she was reading the description on the site's home page. Then she entered the register page and tried to register. She was unable to sign up for the first time because she created a password with less characters than the password character limit. The page was refreshed and the information she entered was deleted. She stated that it would be better if the information she wrote had remained. After opening the home page, she first entered her own profile page. She said that in the section with the number of people the user follows, it would be nice to see who she followed when clicked on it. When she saw my comments section on the profile, she said it was good to reach the comments easily. She was able to easily change her cover and profile photos. She stated that it might be better to limit the cover photo to a certain size. She said that she liked the image section on the right in her profile, but she stated that it would be better to see the big version when clicking on the pictures. When she got back to the home page, she had a hard time sharing posts. She tried to click on the "What do you think?" text without showing any interest in the share button on the side. After realizing that she had to click the share button, she was able to easily post normal posts and posts with picture. She mentioned that she liked the personal information panel on the left of the main page. She commented on a post in the post flow without any problems. She said it would be nice to like a comment and stated that it is a bit annoying if the comments are not deleted directly from the post. She said she would prefer the caption to be above the picture for posts with pictures and when she tried to like that post, she said it looks like it cannot be taken back after she liked it. After the main page, she easily found the settings page and understood the purpose of the functions there and how they are used. She had no trouble logging out of the site. She tried the forgot password function. She stated that the function works actively and nicely. With the information from the forgot password e-mail, she logged into her account again and easily changed her password on the settings page.

As a result, although she had a few difficulties and thought some things could be better , she said she had a good experience and liked the site in general.

# conclusions and future work

In this project, we aimed to make a social media website that will be an alternative to existing social media and has a specific theme. Its purpose is to create a platform that will bring together people who are interested in culinary art and gastronomy or have a profession in the field of gastronomy. It has a user friendly interface that users can easily use.

Our project has the minimum requirements of a social media website. In other words, our website performs the basic functions that can be found on an average social media site, such as sharing, following, commenting, liking. From this perspective, our project has fulfilled the requirements of success criteria in general. In terms of design, our project does not have features that can make the website more dynamic compared to the websites that come to mind when it comes to social media. However, since the current version is the first version of the project, these are acceptable deficiencies and therefore do not negatively affect the success criteria. All of the problems we experienced during the project were the problems caused by not using the web application framework on the frontend of the project. We overcame these problems with the node packages we used in the project. In the following process, the project will first be developed in terms of the technologies used and will have the next version. This will make the site more dynamic.

# References

*Approximate string matching*. 2021. <en.wikipedia.org/wiki/Approximate\_string\_matching>.

*Bcrypt*. 2021. <https://en.wikipedia.org/wiki/Bcrypt>.

Gambolthemes. *The Freelancer And Social Networking HTML Template*. 2019 <themeforest.net/user/gambolthemes>.

Koçoğlu, Sinem. *Instagram Tarihi: Instagram Nedir? Nasıl Kullanılır? Ne İşe Yarar?* 3 March 2018. <brandingturkiye.com>.

Konuk, Ali Haydar and Çoşkun, Yaren. «Requirements Analysis Document.» 2021. <https://github.com/alikonuk12/AKYC/blob/master/Documents/RAD/AKYC\_RAD.docx>.

Konuk, Ali Haydar and Çoşkun, Yaren. «System Design Document.» 2021. <https://github.com/alikonuk12/AKYC/blob/master/Documents/SDD/AKYC\_SDD.docx>.

Rouse, Margaret. *Fuzzy Search*. 2005. <whatis.techtarget.com>.

*What is the Difference Between Twitter and Facebook?* 2016. <https://www.visualscope.com/twitfb.html>.