Table of contents:

[DEFINITION**:** 2](#_Toc5303659)

[Backgroundinformation**:** 2](#_Toc5303660)

[Methods in use: 2](#_Toc5303661)

[Origin of the data: 3](#_Toc5303662)

[INVESTIGATION: 4](#_Toc5303663)

[Interview: 4](#_Toc5303664)

[ANALYSIS: 7](#_Toc5303665)

[Flow chart: 7](#_Toc5303666)

[Data flow diagram: 7](#_Toc5303667)

[Data capture methods: 8](#_Toc5303668)

[Alternative methods: 8](#_Toc5303669)

[REQUIREMENTS: 10](#_Toc5303670)

[User requirements: 10](#_Toc5303671)

[Software requirements: 10](#_Toc5303672)

[Hardware requirements: 11](#_Toc5303673)

[DESIGN: 12](#_Toc5303674)

[Objectives: 12](#_Toc5303675)

[ERDs: 12](#_Toc5303676)

[Prototype: 13](#_Toc5303677)

[Updated prototype: 14](#_Toc5303678)

[Flow chart: 14](#_Toc5303679)

[First level DFD: 15](#_Toc5303680)

[Limitations: 16](#_Toc5303681)

[DEVELOPMENT: 17](#_Toc5303682)

[1. Login page (index.php): 17](#_Toc5303683)

[2. Registration page (reg.php): 18](#_Toc5303684)

[3. Homepage (index.php): 19](#_Toc5303685)

[4. Article page (warticle.php): 23](#_Toc5303686)

[5. Profile page (profile.php): 26](#_Toc5303687)

[6. Profile change page (profile\_change.php): 30](#_Toc5303688)

[7. Email confirm page (emailconfirm.php): 34](#_Toc5303689)

[8. Php file that contains all functional (server.php): 35](#_Toc5303690)

[9. Errors representing file (errors.php): 39](#_Toc5303691)

[TESTING: 40](#_Toc5303692)

[Testing plan: 40](#_Toc5303693)

[System testing: 41](#_Toc5303694)

[Data validation testing table: 49](#_Toc5303695)

[System installation: 52](#_Toc5303696)

[User manual: 52](#_Toc5303697)

[Evaluation: 52](#_Toc5303698)

# **DEFINITION:**

## **Background information:**

The customer is the school council of the Autonomous Organization of Education "Nazarbayev Intellectual School" company located in Almaty region, Taldykorgan that works for 8 years.

Every year there is president election in the school, where teachers, curators, students from 7 to 12 grade and et cetera. President carries a group of people that are dedicated to the posts of ministers, who are respond for four major spheres, such as education, social projects, culture and sports. A school council consists of 8 students. Four of the students were assigned on the post of minister and others are respond for different spheres within the ministry. The main purpose of the school council is to build a bridge between school administration and students.

My customer is a school council’s minister of education, who’s main aim is to conduct different meetings and inform people about them to make students more competent in educational sphere. The problem is that students of high school are unsure of the exams that they need to pass and what do they need to do to get to the university. I was excited of the idea of creating school website for middle school students to dispose of this issue.

## **Methods in use:**

There are some prof oriented meetings like “Smart Thursday” between NIS and different universities’ students or educational centers in the school, where second are explaining the key information about the university itself, exams that you need to pass and features of the country or town, where university is located.

Nowadays students don’t have any connection to each other and younger classes don’t know what to expect in higher grades. The only solution for this situation is to ask high school students orally or by texting a message, but the problem is that middle school students are too shy and originally they don’t even know about the information they may receive.

This information and connection is important for school students not to lose opportunities to get to the university they really want and to start thinking about their future not only in high school, but in younger grades too. It would be important to contact graduated students to ask them to participate and give different advices for school students community.

Also, there is a lamp evening, where graduated students meet other students, where they deal with experience of receiving qualitive education, achieving their goals and reach educational success.

## **Origin of the data:**

Career counselor provides information via personal or public meetings. Students from high school meet middle-school students on self-rule day.

**Form the data takes:**

# **INVESTIGATION:**

The interview method was chosen, to gather information about the work type the organization does, the methods that are currently used and to determine what does consumer expects from such work in the form of extended answers and descriptions. The interviewer is the school council’s minister of education.

## **Interview:**

**Question 1:** What is the main purpose of your organization?

**Answer 1:** Speaking about cabinet of ministers, our main purpose is the organization of school council’s work and what is most important providing connection between students and administration. I mean if students have some wishes or ideas we are trying to make them alive. As minister of education I inform students about olympiads and other school activities.

**Question 2:** What is your target audience?

**Answer 2:** Speaking about the school council the target audience is students. Speaking about me as the minister of education it is students too, since every event and activity is basically aimed on students.

**Question 3:** In which educational activities do you participate?

**Answer 3:** We conduct different meetings with graduated students, representors of different colleges, universities and organizations. These activities are aimed on narrower audience, I mean, only for some grade parallels, but mostly it is preferable and every student may visit any activity he or she wants.

**Question 4:** What are the main problems in organization of this activities?

**Answer 4:** Sometimes we face with difficulties when we can’t gather people, sometimes we can’t announce about meeting at time, because the school schedule is so fast-changing and you don’t even have time to notify students. Sometimes it is hard to work with students, since they usually don’t even know what they want.

**Question 5:** What kind of information do students get during prof oriented meetings?

**Answer 5:** Basically they provide general information about the universities and answer on all questions that they receive from our students.

**Question 6:** Where do you get this information from?

**Answer 6:** Prof oriented meetings are mostly meetings with graduated students, different universities’ representors and organizations, which help to graduate students in foreign countries. When we know that graduated students at the school we ask them to conduct a prof oriented meeting to answer on different students’ questions.

**Question 7:** What kind of meetings do you organize for young students to discover their future perspectives?

**Answer 7:** For example, this year we organized a meeting for seventh, eighth grades, where they were introduced about different popular spheres and professions in their grade. We think that this information is fulfilling for their age.

**Question 8:** How do students from middle school receive information about the exams and universities?

**Answer 8:** We have smart thursdays, where students have meetings with universities representors and there they may receive all needed information.Also we provide self-rule day, where students meet 12th grade students and there they may ask any questions they want.

**Question 9:** How do you connect middle and high school students?

**Answer 9:** First of all we have shanyrak system, I mean we have so many activities where every shanyrak is involved. Secondly, during different meetings we try to engage high grade students to show students from middle school that they can ask something that they want from not only school council but from every student individually.

**Question 10:** Do you think that connection would be useful?

**Answer 10:** No doubt this connection is very useful for high school students and also middle school students, because they can ask questions that they are interested in and can receive answers from people, who have earned such experience.

**Question 11:** Do you think that local social network would be useful to provide such kind of information from high to middle school students?

**Answer 11:** Yes, of course.

**Question 12:** What do you want to to see on the website?

**Answer 12:** Basically, I want to create a stronger connection between students of different grades (middle and high school, also graduates). I want to see how students really help each other and become more that just a school.

**Question 13:** What style of information sharing do you think is the most applicable for this project?

**Answer 13:**I don’t believe that personal messages or something like that will help in this occasions. Maybe blog style will be useful for us. Younger students are going to receive information from reading different articles of higher grade students.

**Question 14:** Do you want students to be able to delete their own articles?

**Answer 14:** No, I think that every information should stay on the website and if an article has inappropriate information I want for only school administration to be able to delete articles. If student wants to delete it from site, he or she may contact us by the information provided in the end of the every page.

**Question 15:** What does the profile of students should contain?

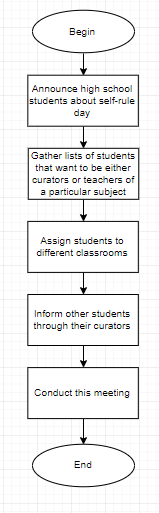
**Answer 15:**Name, surname, grade with a letter, and some additional information like preferences on subjects, hobbies, role at school (responsible for events in the class, president), participating clubs in the school.

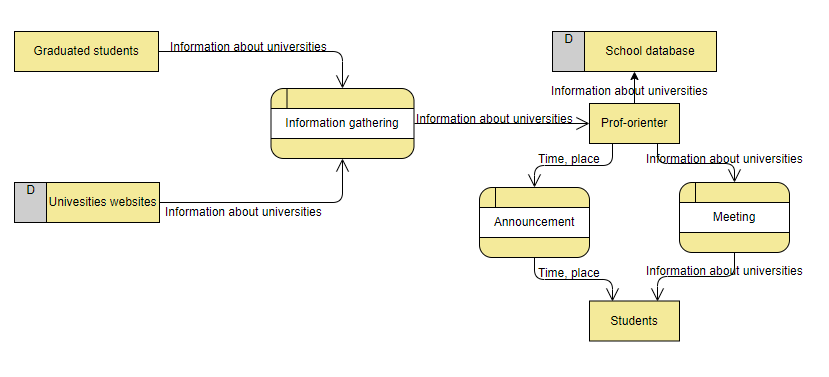
**Question 16:** Do you have any recommendations for the social network?

**Answer 16:**I think that articles have too much information and we need something shorter like advises. You may add this on the home page of the website.

ANALYSIS:

Flow chart:

****

Data flow diagram:****

Data capture methods:

The information about different exams and universities is going to be served for school students from graduated and pre-graduated students. Second are going to write articles that will be shown on the home page, so that first can visit website and read these articles. The data will be saved on the school database. The main feature of this method is that students will receive only actual information and in comfortable conditions (anytime they want). The main con is that there is no 100% guarantee of students’ attention. Engagement of students depends on their interest and quality of the content that pre-graduated and graduated students will produce.

Alternative methods:

Conferences, where graduated and enrollee students will speak for younger grades’ students about their path to the university, all the exams, olympiads and steps that they already passed.

Advantages:

* Students receive only actual information, since they get it from original sources.
* Atmosphere of the meeting helps students to concentrate. Since they sit in big auditory and conferences have official form, the information recieved by students is going to have emotional context.
* Most of the students receive needed information simultaneously. Usually many students are gathered in one room and there is no need of learning each student separately.
* Guarantee that students received data. When students are gathered in a room, you absolutely know that they hear needed information.
* Students may ask any questions they want. After such meetings students usually have opportunity to ask questions students are interested in.

Disadvantages:

* Sometimes students do not have time to visit such meetings.
* Hard to organize, because guests are needed to be invited.
* Such meetings are conducted rarely.

Wix website. Wix is a cloud-based platform for creating and promoting internet-projects that allows to construct websites using drag-and-drop technology.

Advantages:

* Usable and comfortable design for customer.
* It does not require programming skills. System is intuitively understandable for programmer and this environment provides different templates.
* Wix gives opportunities to work on different platforms such as smartphone, pc or tablet.
* Wix provides free hosting.

Disadvantages:

* Stupid domain name. It is hard to find website with long and weird domain name.
* No opportunities to work offline.
* Most of features are not free.
* Low level of flexibility. Programmer can not finally change the design.

Edmodo is a platform for educational purposes that provides communication, collaboration and coaching opportunities.

Advantages:

* Students are familiar with this system.
* Only those who have group code can enter the group.

Disadvantages:

* Any student may post anything.
* Poor visual part of articles representation.

Requirements:

User requirements:

* System should have extended user settings. User should be able to write more information about them, not only their names, grades, and shanyraks, preferences on subjects, hobbies, role at school (responsible for events in the class, president) and participating clubs in the school.
* Website should include activity streams with threaded commenting and direct posting.
* System should include the main feed page with the infinite scrolling system.
* All web pages should be in russian.
* Website should include advices section on the home page.

Software requirements:

|  |  |
| --- | --- |
| **Softwarerequirements:** | **Reason:** |
| Windows Operating System | System to run applications on my computer. |
| MicrosoftWord | Write a documentation about the project. |
| Adobe Photoshop | Logo drawing and photo corrections. |
| MAMP | It is the local server environment to run wordpress on the computer, use PhP and MySQL. |
| Word Press | The platform to create a website. |
| Buddy Pressplugin | The plugin to make a social network with extended profiles, private messages, activity streams, user settings, estensible groups, friend connections. |
| Chrome browser | The browser to use wordpress and see all changes of code. |

Hardware requirements:

|  |  |  |
| --- | --- | --- |
|  | **Device** | **Function** |
| Input devices | Qwerty Keyboard | To input the data. |
|  | Laser mouse | To input the data. |
| Output devices | Flat monitor | To see the result. |
|  | Headphones, loudspeakers. | To output music and sounds from the website. |
| Internal devices | Processor, external hard drive, RAM, cache memory. | To make hardware work, open web browser, run applications. |

DESIGN:

Objectives:

To create a universal navigation bar for every web page with home, articles and logout sections.

To create a sidebar for homepage with profile, advice of the day and last blog sections.

To draw a logo for social network.

To create working registration and login forms.

ERDs:

**Table “user”:**

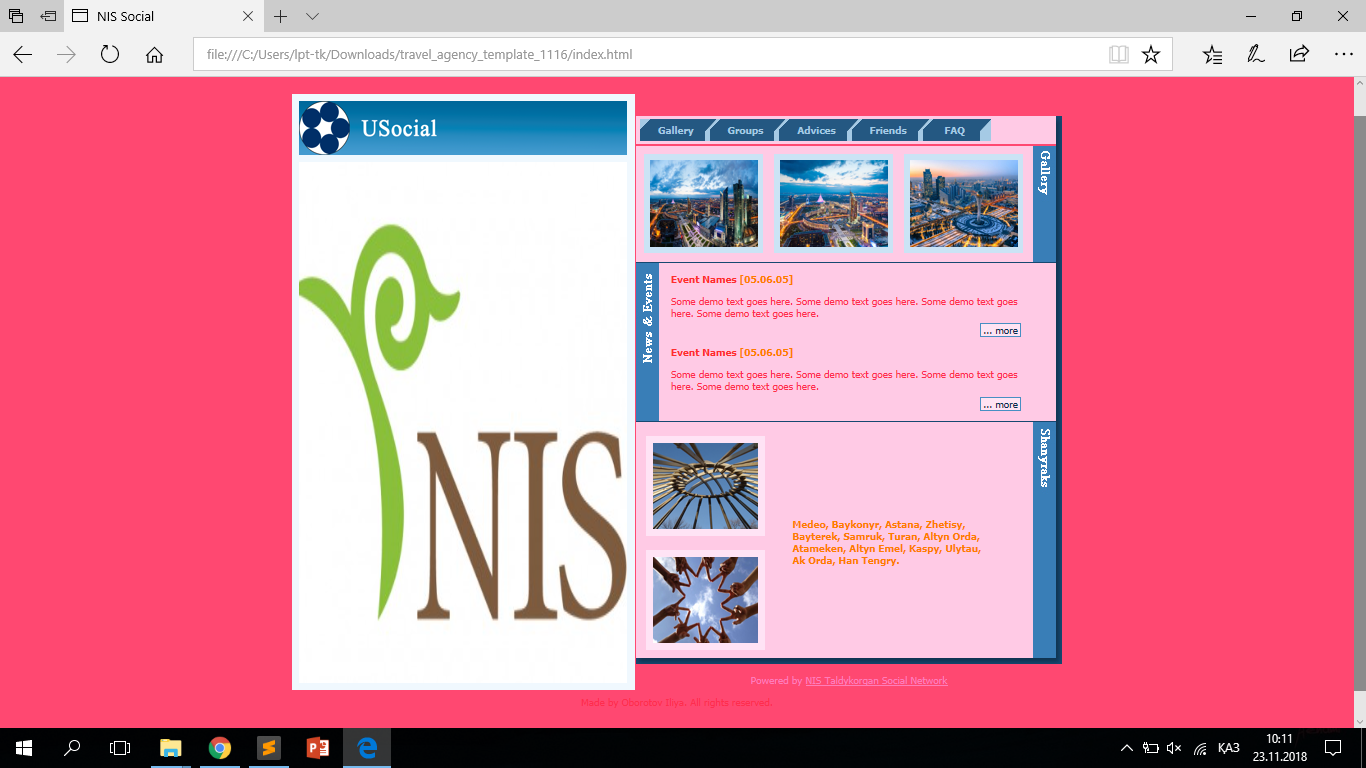
|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Size / symbols** | **Key** |
| id | int | 11 | primary |
| username | varchar | 25 | foreign |
| password | varchar | 25 | - |
| email | varchar | 25 | - |
| fname | varchar | 50 | - |
| grade | varchar | 2 | - |
| interests | text | 63 535 | - |
| litera | varchar | 1 | - |
| confirmed | int | 1 | - |
| ccode | int | 11 | - |

**Table “articles”:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Size / symbols** | **Key** |
| articleid | int | 11 | primary |
| time | datetime | 18 | - |
| content | text | 63 535 | - |
| aname | varchar | 50 | - |
| username | varchar | 25 | foreign |

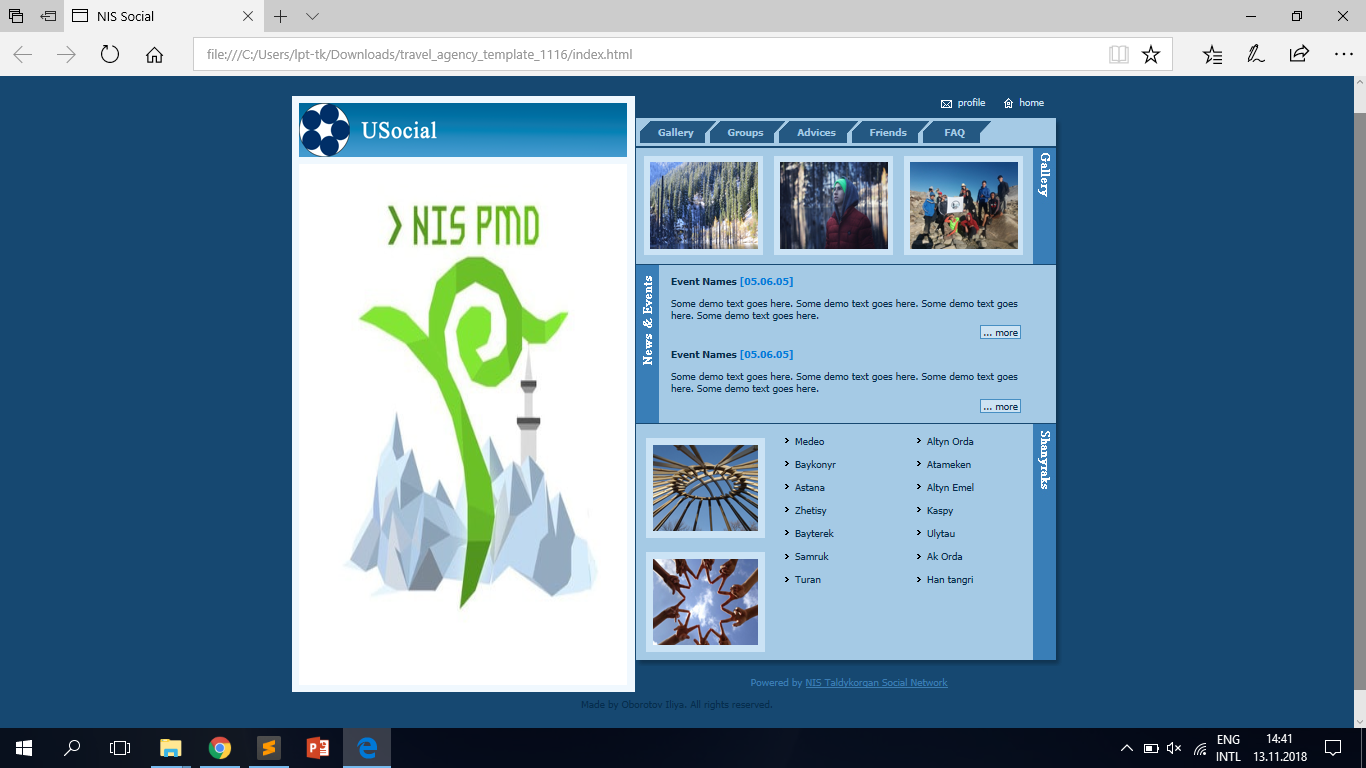
**Table “advices”:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Size / symbols** | **Key** |
| advice\_id | int | 11 | primary |
| content | varchar | 100 | - |

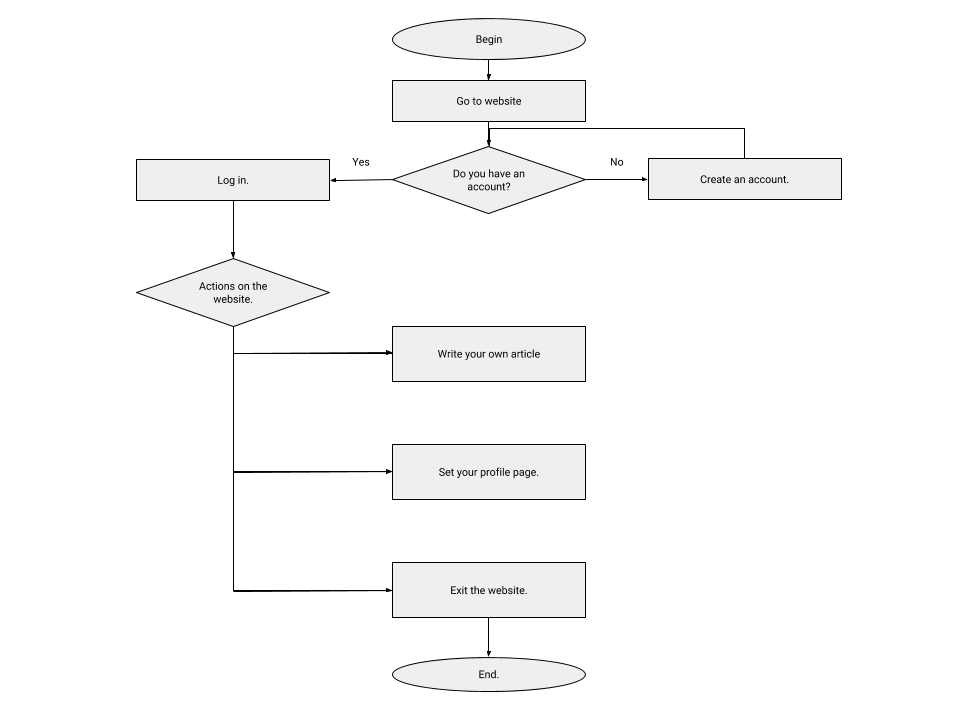
Prototype:

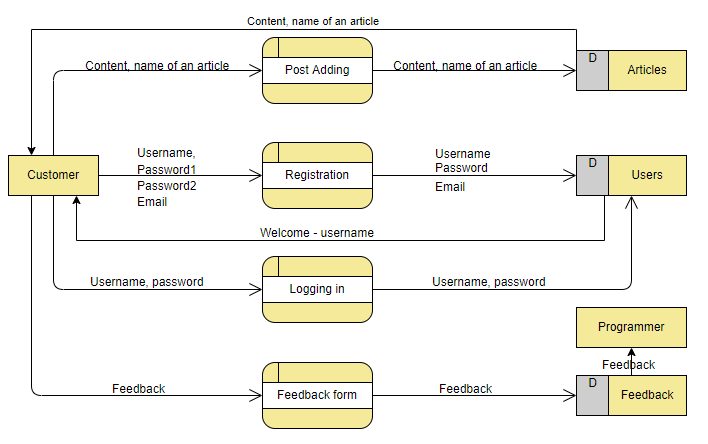
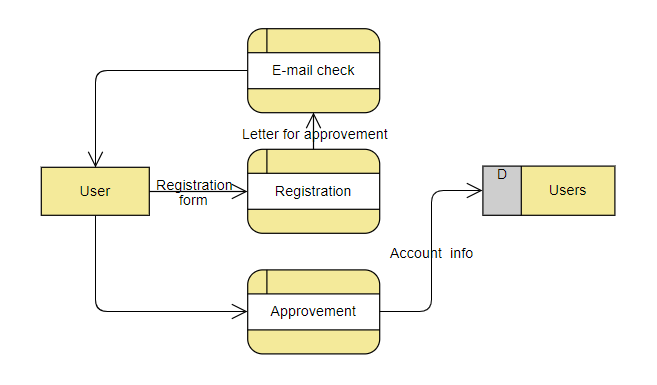
Feedback:

1. Detrimental color choice. Change all red parts of the website to blue, because it already has blue lines, where “Gallery”, “News & Events”, “Shanyraks” and logo of social network stand.
2. Inappropriate logo for the website. It seems unsuitable on this page, so it is needed to replace it with another logo or picture.
3. Add profile and home references to the navigation.
4. Change pictures in the Gallery section. The pictures that are placed in this section doesn’t belong to the NIS, so it could be better to change them to pictures made during activities that was conducted by NIS organisation.

Updated prototype:

Flow chart:

****

First level DFD:****Zero level DFD:****

**Intended benefits:**

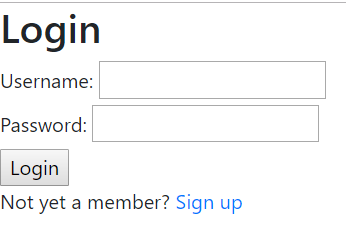
My system will allow to save professional information about universities, students’ personal experience and data about exams and test that students need to perform. My system provides only the most actual information due to pre-graduated and graduated students are going to deal with current problems and achievements. My system is more comfortable, because it doesn’t depend on time, since any student may visit website any time they want.

Limitations:

1. Information overwhelm. Big amount of records, videos, photos, profiles require large memory storage to keep these data. Every customer of the social network creates or shares the data, so the website with many users need a lot of space to store all information.
2. Distraction and procrastination. Usually people get distracted by messages or records that are not related to current situation, so whilst the customer tries to find information about particular exams that he needs to pass, the information about different universities may be distractive.

Delevopment:

1. Login page (index.php):

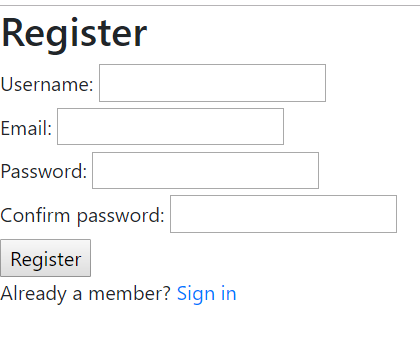


This page is made to enter the profile.

1. <?**php**
2. include('server.php');
3. ?>

6. <!DOCTYPE html>
7. <**html**>
8. <**head**>
9. <**title**>Logging in</**title**>
10. <**link** rel="stylesheet" type="text/css" href="css/bootstrap.css">
11. </**head**>
12. <**body**>
13. <**div**>
14. <**h2**>Login</**h2**>
15. </**div**>
17. <**form** method="post" action="login.php">
18. <?**php**
19. include('errors.php');
20. ?>
21. <**div**>
22. <**label**>Username: </**label**>
23. <**input** type="text" name="username" autocomplete="off">
24. </**div**>
25. <**div**>
26. <**label**>Password: </**label**>
27. <**input** type="password" name="password" autocomplete="off">
28. </**div**>
29. <**div**>
30. <**button** type="submit" name="login">Login</**button**>
31. </**div**>
32. <**p**>Not yet a member? <**a** href="reg.php">Sign up</**a**></**p**>
33. </**form**>
34. </**body**>
35. </**html**>

2. Registration page (reg.php):

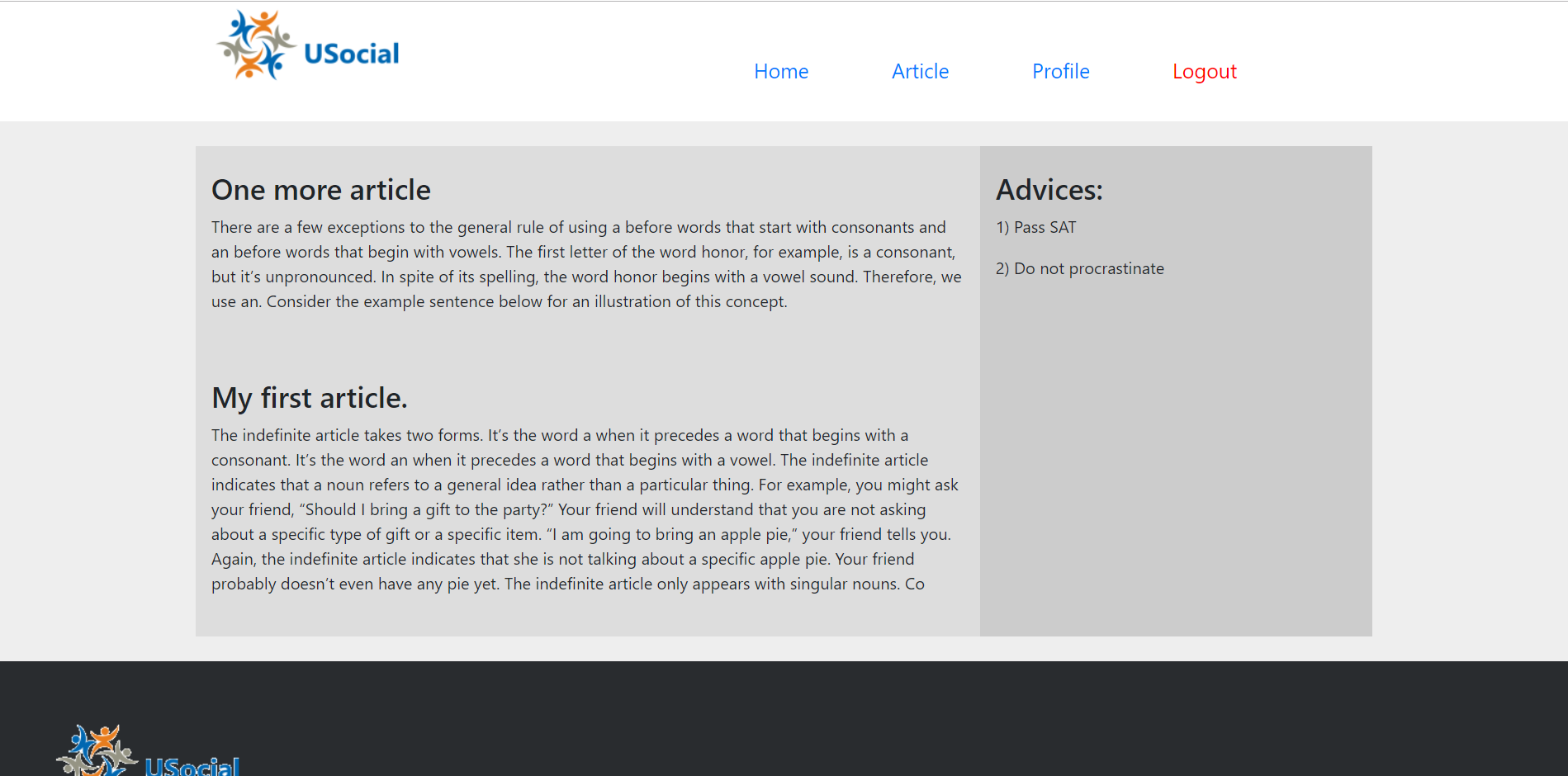


A page to register into the system.

1. <?**php** include('server.php'); ?>

4. <!DOCTYPE html>
5. <**html**>
6. <**head**>
7. <**title**>Registration</**title**>
8. <**link** rel="stylesheet" type="text/css" href="css/bootstrap.css">
9. </**head**>
10. <**body**>
11. <**div**>
12. <**h2**>Register</**h2**>
13. </**div**>
15. <**form** method="post" action="reg.php">
16. <?**php**
17. include('includes/errors.php');
18. ?>
19. <**div**>
20. <**label**>Username: </**label**>
21. <**input** type="text" name="username" autocomplete="off">
22. </**div**>
23. <**div**>
24. <**label**>Email: </**label**>
25. <**input** type="text" name="email" autocomplete="off">
26. </**div**>
27. <**div**>
28. <**label**>Password: </**label**>
29. <**input** type="password" name="password\_1" autocomplete="off">
30. </**div**>
31. <**div**>
32. <**label**>Confirm password: </**label**>
33. <**input** type="password" name="password\_2" autocomplete="off">
34. </**div**>
35. <**div**>
36. <**button** type="submit" name="register">Register</**button**>
37. </**div**>
38. <**p**>Already a member? <**a** href="login.php">Sign in</**a**></**p**>
39. </**form**>
40. </**body**>
41. </**html**>

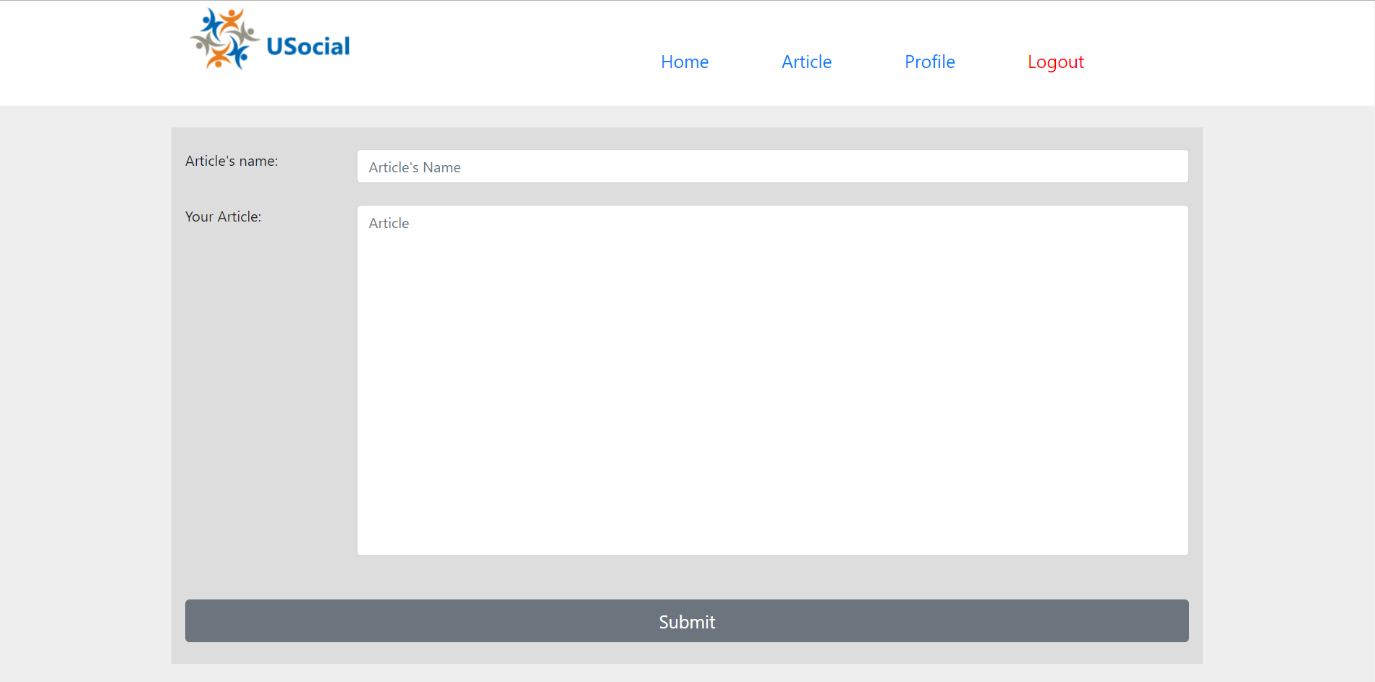
3. Homepage (index.php):



To use the website user firstly needs to register of log in. Main page contains of navbar, infinite tape of articles, advices and footer.

1. <?**php** include('server.php');
3. if (empty($\_SESSION['username'])) {
4. header('location: login.php');
5. }
6. if (isset($\_GET['logout'])) {
7. session\_destroy();
8. unset($\_SESSION['username']);
9. header("location: login.php");
10. }
11. $db = mysqli\_connect('localhost', 'root', '', 'usocial');
12. ?>
14. <**html**>
15. <**head**>
16. <**meta** charset="utf-8">
17. <**meta** name="viewport" content="width=device-width, install-scale=1">
18. <**link** rel="stylesheet" type="text/css" href="css/bootstrap.css">
19. <**title**>USocial|Коннект</**title**>
20. </**head**>
21. <**body**>
22. <**div** style="">
23. <**div** class="container" style="margin-top: 0px;">
24. <**div** class="row">
25. <**div** class="col-md-4">
26. <**a** href="#"> <**img** src="img/Logo.png"> </**a**>
27. </**div**>
28. <**div** class="col-md-8" id="topnav">
29. <**ul** class="col-md-11">
30. <**li**>
31. <?**php** if(isset($\_SESSION['username'])): ?>
32. <**p**><**a** href="index.php?logout='1'" style="color: red;">Logout</**a**></**p**>
33. <?**php** endif ?>
34. </**li**>
35. <**li**>
36. <**a** href="profile.php">Profile</**a**>
37. </**li**>
38. <**li**>
39. <**a** href="warticle.php">Article</**a**>
40. </**li**>
41. <**li**>
42. <**a** href="index.php">Home</**a**>
43. </**li**>
44. </**ul**>
45. </**div**>
46. </**div**>
47. </**div**>
48. </**div**>
49. <**div** style="background-color: #eeeeee">
50. <**div** class="container">
51. <**br**>
52. <**div** class="row">
53. <**div** class="col-md-8" style="background-color: #dddddd;">
54. <?**php**
55. $sql = "SELECT aname, content FROM articles ORDER BY time DESC";
56. $result = $db->query($sql);
57. if ($result->num\_rows > 0) {
58. while ($row = $result->fetch\_assoc()) {
59. echo "<**br**>"."<**h3**>",$row['aname'],"</**h3**>"."<**p**>",$row['content'],"</**p**>"."<**br**>" ;
60. }
61. } else {
62. echo "0 results";
63. }
64. ?>
65. </**div**>
66. <**div** class="col-md-4" style="background-color: #cccccc;">
67. <**br**>
68. <**h3** >Advices:</**h3**>
69. <?**php**
70. $sql = "SELECT advice\_id, content FROM advices";
71. $result = $db->query($sql);
72. if ($result->num\_rows > 0) {
73. while ($row = $result->fetch\_assoc()) {
74. echo "<**p**>",$row['advice\_id'],") ",$row['content'],"</**p**>";
75. }
76. } else {
77. echo "0 results";
78. }
79. ?>
80. </**div**>
81. </**div**>
82. </**div**>
83. <**br**>
84. </**div**>
86. <**footer** class="footer-distributed" style="margin-top: 0px;>
87. <**div** class="footer-left">
89. <**h3**><**img** src="img/FootLogo.png"></**h3**>
91. <**p** class="footer-links">
92. <**a** href="#">Home</**a**>
93. ·
94. <**a** href="#">Articles</**a**>
95. ·
96. <**a** href="#">About</**a**>
97. ·
98. <**a** href="#">Faq</**a**>
99. ·
100. <**a** href="#">Contact</**a**>
101. </**p**>
103. <**p** class="footer-company-name">Usocial &copy; 2018</**p**>
104. </**div**>
106. <**div** class="footer-center">
108. <**div**>
109. <**i** class="fa fa-map-marker"></**i**>
110. <**p**><**span**>Abai Street</**span**> Almaty, Kazakhstan</**p**>
111. </**div**>
113. <**div**>
114. <**i** class="fa fa-phone"></**i**>
115. <**p**>+8 707 123 13 00</**p**>
116. </**div**>
118. <**div**>
119. <**i** class="fa fa-envelope"></**i**>
120. <**p**><**a** href="mailto:support@company.com">oborotov\_i@gmail.com</**a**></**p**>
121. </**div**>
123. </**div**>
125. <**div** class = "container">
127. </**div**>
129. <**div** class="footer-right">
131. <**p** class="footer-company-about">
132. <**span**>About the company</**span**>
133. This is the local social network for NIS Taldykogan organisation. Aimed to provide connection between students to increase the level of their awareness
134. </**p**>
136. <**div** class="footer-icons">
138. <**a** href="#"><**i** class="fa fa-facebook"></**i**></**a**>
139. <**a** href="#"><**i** class="fa fa-twitter"></**i**></**a**>
140. <**a** href="#"><**i** class="fa fa-linkedin"></**i**></**a**>
141. <**a** href="#"><**i** class="fa fa-github"></**i**></**a**>
143. </**div**>
145. </**div**>
147. </**footer**>
148. </**body**>
149. </**html**>

4. Article page (warticle.php):

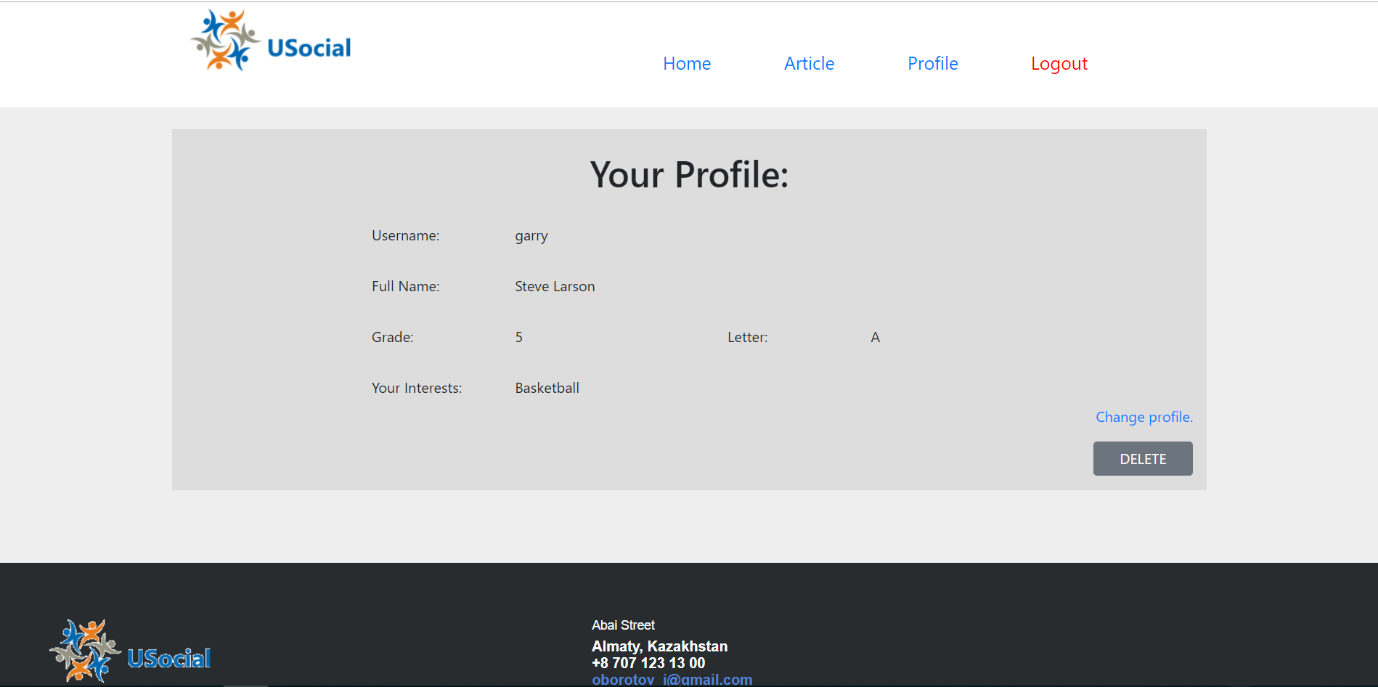


This page is for article posting. Any user can post an article.

1. <?**php** include('server.php');
3. if (empty($\_SESSION['username'])) {
4. header('location: login.php');
5. }
6. if (isset($\_GET['logout'])) {
7. session\_destroy();
8. unset($\_SESSION['username']);
9. header("location: login.php");
10. }
11. ?>
13. <**html**>
14. <**head**>
15. <**meta** charset="utf-8">
16. <**meta** name="viewport" content="width=device-width, install-scale=1">
17. <**link** rel="stylesheet" type="text/css" href="css/bootstrap.css">
18. <**title**>USocial|Коннект</**title**>
19. </**head**>
20. <**body**>
21. <**div** class="container">
22. <**div** class="row">
23. <**div** class="col-md-4">
24. <**a** href="index.php"> <**img** src="img/Logo.png"> </**a**>
25. </**div**>
26. <**div** class="col-md-8" id="topnav">
27. <**ul** class="col-md-11">
28. <**li**>
29. <?**php** if(isset($\_SESSION['username'])): ?>
30. <**p**><**a** href="index.php?logout='1'" style="color: red;">Logout</**a**></**p**>
31. <?**php** endif ?>
32. </**li**>
33. <**li**>
34. <**a** href="profile.php">Profile</**a**>
35. </**li**>
36. <**li**>
37. <**a** href="warticle.php">Article</**a**>
38. </**li**>
39. <**li**>
40. <**a** href="index.php">Home</**a**>
41. </**li**>
42. </**ul**>
43. </**div**>
44. </**div**>
45. </**div**>

48. <**div** style="background-color: #eeeeee">
49. <**br**>
50. <**div** class="container warticle" style="background-color: #dddddd;">
51. <**br**>
52. <**form** action="warticle.php" method="POST">
53. <**div** class="text-center" style="color: red; text-decoration: underline; "><?**php**
54. include('errors.php');
55. ?></**div**>
56. <**div** class = "row">
57. <**div** class="col-md-2">
58. <**label**>Article's name:</**label**>
59. </**div**>
60. <**div** class="col-md-10">
61. <**input** type="text" class="form-control" name="aname" placeholder="Article's Name" autocomplete="off">
62. </**div**>
63. </**div**>
64. <**br**>
65. <**div** class = "row">
66. <**div** class="col-md-2">
67. <**label**>Your Article:</**label**>
68. </**div**>
69. <**div** class="col-md-10">
70. <**textarea** class="form-control" autocomplete="off" wrap="off" name="content" placeholder="Article" rows="15" style="resize: none; outline: none;"></**textarea**>
71. </**div**>
72. </**div**>
73. <**br**>
74. <**br**>
75. <**button** class="btn btn-lg btn-block btn-secondary" type="submit" name = "apost">  Submit  </**button**>
76. <**br**>
77. </**form**>
78. </**div**>
79. <**br**>
80. </**div**>
82. <**footer** class="footer-distributed">
84. <**div** class="footer-left">
86. <**h3**><**img** src="img/FootLogo.png"></**h3**>
88. <**p** class="footer-links">
89. <**a** href="#">Home</**a**>
90. ·
91. <**a** href="#">Articles</**a**>
92. ·
93. <**a** href="#">About</**a**>
94. ·
95. <**a** href="#">Faq</**a**>
96. ·
97. <**a** href="#">Contact</**a**>
98. </**p**>
100. <**p** class="footer-company-name">Usocial &copy; 2018</**p**>
101. </**div**>
103. <**div** class="footer-center">
105. <**div**>
106. <**i** class="fa fa-map-marker"></**i**>
107. <**p**><**span**>Abai Street</**span**> Almaty, Kazakhstan</**p**>
108. </**div**>
110. <**div**>
111. <**i** class="fa fa-phone"></**i**>
112. <**p**>+8 707 123 13 00</**p**>
113. </**div**>
115. <**div**>
116. <**i** class="fa fa-envelope"></**i**>
117. <**p**><**a** href="mailto:support@company.com">oborotov\_i@gmail.com</**a**></**p**>
118. </**div**>
120. </**div**>
122. <**div** class = "container">
124. </**div**>
126. <**div** class="footer-right">
128. <**p** class="footer-company-about">
129. <**span**>About the company</**span**>
130. This is the local social network for NIS Taldykogan organisation. Aimed to provide connection between students to increase the level of their awareness
131. </**p**>
133. <**div** class="footer-icons">
135. <**a** href="#"><**i** class="fa fa-facebook"></**i**></**a**>
136. <**a** href="#"><**i** class="fa fa-twitter"></**i**></**a**>
137. <**a** href="#"><**i** class="fa fa-linkedin"></**i**></**a**>
138. <**a** href="#"><**i** class="fa fa-github"></**i**></**a**>
140. </**div**>
142. </**div**>
144. </**footer**>
145. </**body**>
146. </**html**>

5. Profile page (profile.php):

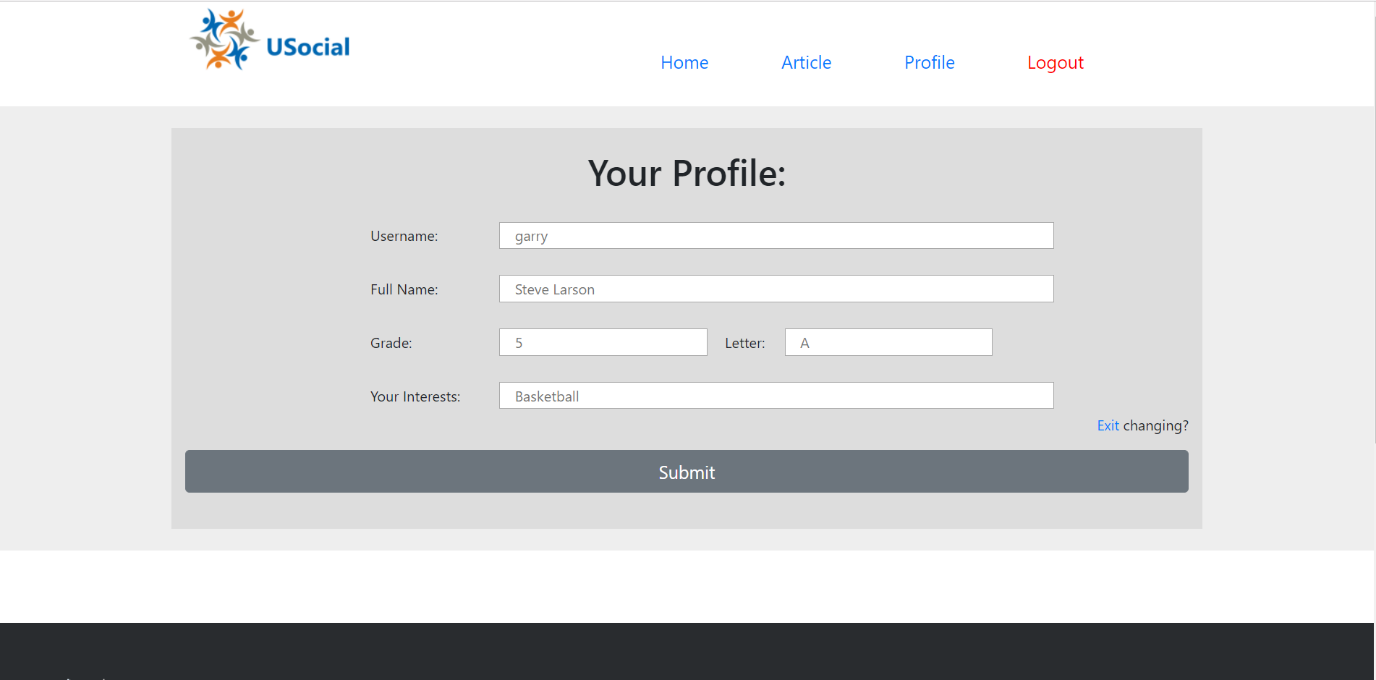


On this page user may see the information of his profile.

1. <?**php** include('server.php');
3. if (empty($\_SESSION['username'])) {
4. header('location: login.php');
5. }
6. if (isset($\_GET['logout'])) {
7. session\_destroy();
8. unset($\_SESSION['username']);
9. header("location: login.php");
10. }
11. $db = mysqli\_connect('localhost', 'root', '', 'usocial');
12. $username = $\_SESSION['username'];
13. ?>
15. <**html**>
16. <**head**>
17. <**meta** charset="utf-8">
18. <**meta** name="viewport" content="width=device-width, install-scale=1">
19. <**link** rel="stylesheet" type="text/css" href="css/bootstrap.css">
20. <**title**>USocial|Коннект</**title**>
21. </**head**>
22. <**body**>
23. <**div** class="container">
24. <**div** class="row">
25. <**div** class="col-md-4">
26. <**a** href="index.php"> <**img** src="img/Logo.png"> </**a**>
27. </**div**>
28. <**div** class="col-md-8" id="topnav">
29. <**ul** class="col-md-11">
30. <**li**>
31. <?**php** if(isset($\_SESSION['username'])): ?>
32. <**p**><**a** href="index.php?logout='1'" style="color: red;">Logout</**a**></**p**>
33. <?**php** endif ?>
34. </**li**>
35. <**li**>
36. <**a** href="profile.php">Profile</**a**>
37. </**li**>
38. <**li**>
39. <**a** href="warticle.php">Article</**a**>
40. </**li**>
41. <**li**>
42. <**a** href="index.php">Home</**a**>
43. </**li**>
44. </**ul**>
45. </**div**>
46. </**div**>
47. </**div**>
48. <**div** style="background-color: #eeeeee">
49. <**br**>
50. <**div** class="container" style="background-color: #dddddd;">
51. <**br**>
52. <**div** class="text-center">
53. <**h1**>Your Profile:</**h1**>
54. </**div**>
55. <**div** class="row">
56. <?**php**
57. include('errors.php');
58. ?>
59. <**div** class="col-md-2">
60. <**img** src="">
61. </**div**>
62. <**div** class="col-md-10">
63. <**br**>
64. <**div**>
65. <**label** class="col-md-2">Username:</**label**>
66. <**label** class="col-md-8"><?**php** echo $username ?></**label**>
67. </**div**>
68. <**br**>
69. <**div**>
70. <**label** class="col-md-2">Full Name:</**label**>
71. <**label** class="col-md-8"><?**php** $sql = "SELECT fname FROM user WHERE username = '$username';"; $result = $db->query($sql); if(!empty($result)) { $data = mysqli\_fetch\_array($result); echo $data[0]; } ?></**label**>
72. </**div**>
73. <**br**>
74. <**div**>
75. <**label** class="col-md-2">Grade:</**label**>
76. <**label** class="col-md-3"><?**php** $sql = "SELECT grade FROM user WHERE username = '$username';"; $result = $db->query($sql); if(!empty($result)) { $data = mysqli\_fetch\_array($result); echo $data[0]; } ?></**label**>
77. <**label** class="col-md-2">Letter:</**label**>
78. <**label** class="col-md-3"><?**php** $sql = "SELECT litera FROM user WHERE username = '$username';"; $result = $db->query($sql); if(!empty($result)) { $data = mysqli\_fetch\_array($result); echo $data[0]; } ?></**label**>
79. </**div**>
80. <**br**>
81. <**div**>
82. <**label** class="col-md-2">Your Interests:</**label**>
83. <**label** class="col-md-8"><?**php** $sql = "SELECT interests FROM user WHERE username = '$username';"; $result = $db->query($sql); if(!empty($result)) { $data = mysqli\_fetch\_array($result); echo $data[0]; } ?></**label**>
84. </**div**>
85. <**div** class="text-right">
86. <**p**><**a** href="profile\_change.php">Change profile.</**a**></**p**>
87. <**form** action="server.php" method="POST"><**button** class="btn btn-secondary" style="width: 110px;" type="submit" name="delete">DELETE</**button**></**form**>
88. </**div**>
89. </**div**>
90. </**div**>
91. </**div**>

94. <**footer** class="footer-distributed">
96. <**div** class="footer-left">
98. <**h3**><**img** src="img/FootLogo.png"></**h3**>
100. <**p** class="footer-links">
101. <**a** href="#">Home</**a**>
102. ·
103. <**a** href="#">Articles</**a**>
104. ·
105. <**a** href="#">About</**a**>
106. ·
107. <**a** href="#">Faq</**a**>
108. ·
109. <**a** href="#">Contact</**a**>
110. </**p**>
112. <**p** class="footer-company-name">Usocial &copy; 2018</**p**>
113. </**div**>
115. <**div** class="footer-center">
117. <**div**>
118. <**i** class="fa fa-map-marker"></**i**>
119. <**p**><**span**>Abai Street</**span**> Almaty, Kazakhstan</**p**>
120. </**div**>
122. <**div**>
123. <**i** class="fa fa-phone"></**i**>
124. <**p**>+8 707 123 13 00</**p**>
125. </**div**>
127. <**div**>
128. <**i** class="fa fa-envelope"></**i**>
129. <**p**><**a** href="mailto:support@company.com">oborotov\_i@gmail.com</**a**></**p**>
130. </**div**>
132. </**div**>
134. <**div** class = "container">
136. </**div**>
138. <**div** class="footer-right">
140. <**p** class="footer-company-about">
141. <**span**>About the company</**span**>
142. This is the local social network for NIS Taldykogan organisation. Aimed to provide connection between students to increase the level of their awareness
143. </**p**>
145. <**div** class="footer-icons">
147. <**a** href="#"><**i** class="fa fa-facebook"></**i**></**a**>
148. <**a** href="#"><**i** class="fa fa-twitter"></**i**></**a**>
149. <**a** href="#"><**i** class="fa fa-linkedin"></**i**></**a**>
150. <**a** href="#"><**i** class="fa fa-github"></**i**></**a**>
152. </**div**>
154. </**div**>
156. </**footer**>
157. </**body**>
158. </**html**>

6. Profile change page (profile\_change.php):



On this page users may change information about them.

1. <?**php** include('server.php');
3. if (empty($\_SESSION['username'])) {
4. header('location: login.php');
5. }
6. if (isset($\_GET['logout'])) {
7. session\_destroy();
8. unset($\_SESSION['username']);
9. header("location: login.php");
10. }
11. $db = mysqli\_connect('localhost', 'root', '', 'usocial');
12. $username = $\_SESSION['username'];
13. ?>
15. <**html**>
16. <**head**>
17. <**meta** charset="utf-8">
18. <**meta** name="viewport" content="width=device-width, install-scale=1">
19. <**link** rel="stylesheet" type="text/css" href="css/bootstrap.css">
20. <**title**>USocial|Коннект</**title**>
21. </**head**>
22. <**body**>
23. <**div** class="container">
24. <**div** class="row">
25. <**div** class="col-md-4">
26. <**a** href="index.php"> <**img** src="img/Logo.png"> </**a**>
27. </**div**>
28. <**div** class="col-md-8" id="topnav">
29. <**ul** class="col-md-11">
30. <**li**>
31. <?**php** if(isset($\_SESSION['username'])): ?>
32. <**p**><**a** href="index.php?logout='1'" style="color: red;">Logout</**a**></**p**>
33. <?**php** endif ?>
34. </**li**>
35. <**li**>
36. <**a** href="profile.php">Profile</**a**>
37. </**li**>
38. <**li**>
39. <**a** href="warticle.php">Article</**a**>
40. </**li**>
41. <**li**>
42. <**a** href="index.php">Home</**a**>
43. </**li**>
44. </**ul**>
45. </**div**>
46. </**div**>
47. </**div**>
49. <**div** style="background-color: #eeeeee">
50. <**br**>
51. <**div** class="container" style="background-color: #dddddd;">
52. <**br**>
53. <**div** class="text-center">
54. <**h1**>Your Profile:</**h1**>
55. </**div**>
56. <**form** action="server.php" method="POST">
57. <**div** class="row">
58. <**div** class="col-md-2"></**div**>
59. <**div** class="col-md-10">
60. <**br**>
61. <**div**>
62. <**label** class="col-md-2">Username:</**label**>
63. <**input** type="text" class="col-md-8" name="uname" autocomplete="off" placeholder="<?php echo $username; ?>">
64. </**div**>
65. <**br**>
66. <**div**>
67. <**label** class="col-md-2">Full Name:</**label**>
68. <**input** type="text" class="col-md-8" autocomplete="off" name="fname" placeholder="<?php $sql = "SELECT fname FROM user WHERE username = '$username';"; $result = $db->query($sql); if(!empty($result)) { $data = mysqli\_fetch\_array($result); echo $data[0]; } ?>">
69. </**div**>
70. <**br**>
71. <**div**>
72. <**label** class="col-md-2">Grade:</**label**>
73. <**input** type="text" class="col-md-3" name="grade" autocomplete="off" placeholder="<?php $sql = "SELECT grade FROM user WHERE username = '$username';"; $result = $db->query($sql); if(!empty($result)) { $data = mysqli\_fetch\_array($result); echo $data[0]; } ?>">
74. <**label** class="col-md-1">Letter:</**label**>
75. <**input** type="text" class="col-3" name="litera" autocomplete="off" placeholder="<?php $sql = "SELECT litera FROM user WHERE username = '$username';"; $result = $db->query($sql); if(!empty($result)) { $data = mysqli\_fetch\_array($result); echo $data[0]; } ?>">
76. </**div**>
77. <**br**>
78. <**div**>
79. <**label** class="col-md-2">Your Interests:</**label**>
80. <**input** type="text" class="col-md-8" name="interests" autocomplete="off" placeholder="<?php $sql = "SELECT interests FROM user WHERE username = '$username';"; $result = $db->query($sql); if(!empty($result)) { $data = mysqli\_fetch\_array($result); echo $data[0]; } ?>">
81. </**div**>
82. <**div** class="text-right">
83. <**p**><**a** href="profile.php">Exit</**a**> changing?</**p**>
84. </**div**>
85. </**div**>
86. </**div**>
87. <**button** class="btn btn-lg btn-block btn-secondary" type="submit" name = "change">  Submit  </**button**>
88. </**form**>
89. <**br**>
90. </**div**>
91. <**br**>
92. </**div**>

95. <**footer** class="footer-distributed">
97. <**div** class="footer-left">
99. <**h3**><**img** src="img/FootLogo.png"></**h3**>
101. <**p** class="footer-links">
102. <**a** href="#">Home</**a**>
103. ·
104. <**a** href="#">Articles</**a**>
105. ·
106. <**a** href="#">About</**a**>
107. ·
108. <**a** href="#">Faq</**a**>
109. ·
110. <**a** href="#">Contact</**a**>
111. </**p**>
113. <**p** class="footer-company-name">Usocial &copy; 2018</**p**>
114. </**div**>
116. <**div** class="footer-center">
118. <**div**>
119. <**i** class="fa fa-map-marker"></**i**>
120. <**p**><**span**>Abai Street</**span**> Almaty, Kazakhstan</**p**>
121. </**div**>
123. <**div**>
124. <**i** class="fa fa-phone"></**i**>
125. <**p**>+8 707 123 13 00</**p**>
126. </**div**>
128. <**div**>
129. <**i** class="fa fa-envelope"></**i**>
130. <**p**><**a** href="mailto:support@company.com">oborotov\_i@gmail.com</**a**></**p**>
131. </**div**>
133. </**div**>
135. <**div** class = "container">
137. </**div**>
139. <**div** class="footer-right">
141. <**p** class="footer-company-about">
142. <**span**>About the company</**span**>
143. This is the local social network for NIS Taldykogan organisation. Aimed to provide connection between students to increase the level of their awareness
144. </**p**>
146. <**div** class="footer-icons">
148. <**a** href="#"><**i** class="fa fa-facebook"></**i**></**a**>
149. <**a** href="#"><**i** class="fa fa-twitter"></**i**></**a**>
150. <**a** href="#"><**i** class="fa fa-linkedin"></**i**></**a**>
151. <**a** href="#"><**i** class="fa fa-github"></**i**></**a**>
153. </**div**>
155. </**div**>
157. </**footer**>
158. </**body**>
159. </**html**>

7. Email confirm page (emailconfirm.php):

If registration is successful, on the page “Registration finished. You may now log in.” label occurs. In order if something is wrong with confirm code or username, page outputs “Username and code do not match” message.

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <title>USocial | Email confirm</title>
5. </head>
6. <body>
7. <?php
9. **include**('server.php');
11. $username = $\_GET['username'];
12. $code = $\_GET['code'];
14. $sql = "SELECT \* FROM user WHERE username = '$username';";
15. $result = $db->query($sql);
16. $data = mysqli\_fetch\_array($result);
17. $dbcode = $data['ccode'];
19. **if** ($code == $dbcode) {
20. $sql = "UPDATE user SET confirmed = '1';";
21. $result = $db->query($sql);
22. $sql = "UPDATE user SET ccode = '0';";
23. $result = $db->query($sql);
25. echo "Registration finished. You may now log in.";
26. } **else** {
27. echo "Username and code do not match";
28. }
29. ?>
30. </body>
31. </html>

8. Php file that contains all functional (server.php):

1. <?php
2. $errors = **array**();
3. session\_start();
5. $db = mysqli\_connect('localhost', 'root', '', 'usocial');
7. **if** (isset($\_POST['register'])) {
8. $username = $\_POST['username'];
9. $email = $\_POST['email'];
10. $password\_1 = $\_POST['password\_1'];
11. $password\_2 = $\_POST['password\_2'];
13. **if** (empty($username)) {
14. array\_push($errors, "Username is required");
15. }
16. **if** ((!empty($username)) && (($username[0] < 'a') || ($username[0] > 'z'))) {
17. array\_push($errors, "Invalid username");
18. }
19. **if** (empty($email)) {
20. array\_push($errors, "Email is required");
21. }
23. $nosymlength = 0; $dogpos = 0;
25. **for** ($i = 0; $i < strlen($email); $i++) {
26. **if** ($email[$i] != '@') {
27. $nosymlength++;
28. }
29. **if** ($email[$i] == '@') {
30. $dogpos = $i;
31. }
32. }
33. **if** ((!empty($email)) && ((($nosymlength + 1) != strlen($email)) || ($email[strlen($email) - 1] == '@') || (($email[0] < 'a') || ($email[0] > 'z')))) {
34. array\_push($errors, "Invalid e-mail");
35. }
37. **if** (!empty($username)) {
38. $sql = "SELECT username FROM user WHERE username = '$username';";
39. $result = $db->query($sql);
40. $data = mysqli\_fetch\_array($result);
42. **if**($data['username'] == $username) {
43. array\_push($errors, "This username is already used");
44. }
45. }
47. **if** (!empty($email)) {
48. $sql = "SELECT email FROM user WHERE email = '$email';";
49. $result = $db->query($sql);
50. $data = mysqli\_fetch\_array($result);
52. **if**($data['email'] == $email) {
53. array\_push($errors, "This email is already used");
54. }
55. }
57. **if** (empty($password\_1)) {
58. array\_push($errors, "Password is required");
59. } **else** **if** (strlen($password\_1) > 15) {
60. array\_push($errors, "Password is too long");
61. } **else** **if** (($password\_1) < 6) {
62. array\_push($errors, "Password is too short");
63. }
65. **if** ($password\_1 != $password\_2) {
66. array\_push($errors, "Passwords do not match");
67. }
69. **if** (count($errors) == 0) {
70. $ccode = rand();
72. $sql = "INSERT INTO user (username, email, password, ccode, confirmed) VALUES ('$username', '$email', '$password\_1', '$ccode', '0');"; // password без $
73. mysqli\_query($db, $sql);
75. $message =
76. "
77. Good day!
78. You have registered on USocial website. To finish the registration, please confirm your e-mail via this link:
79. <http://localhost/USocial/emailconfirm.php?username=>$username&code=$ccode
80. ";
82. mail($email, "USocial confirm email", $message);
84. echo "Registration complete. Please confirm your e-mail.";
85. }
86. }
88. **if**(isset($\_POST['login'])) {
89. $username = $\_POST['username'];
90. $password = $\_POST['password'];
91. $sql = "SELECT confirmed FROM user WHERE username = '$username'";
92. $result = $db->query($sql);
93. $data = mysqli\_fetch\_array($result);
95. **if** (empty($username)) {
96. array\_push($errors, "Username is required");
97. }
98. **if** (empty($password)) {
99. array\_push($errors, "Password is required");
100. }
101. **if** ($data['confirmed'] != 1) {
102. array\_push($errors, "Email is not confirmed");
103. }
105. **if** (count($errors) == 0) {
106. $sql = "SELECT \* FROM user WHERE username = '$username' AND password = '$password'"; //здесь было $password = '$password' и users, a не user
107. $result = $db->query($sql);
108. **if** (mysqli\_num\_rows($result) == 1) {
109. $\_SESSION['username'] = $username;
110. $\_SESSION['success'] = "You are now logged in";
111. header('location: index.php');
112. } **else** {
113. array\_push($errors, "Wrong username/password combination");
114. header('location: login.php');
115. }
116. }
117. }
119. **if** (isset($\_POST['apost'])) {
120. $aname = $\_POST['aname'];
121. $content = $\_POST['content'];
122. $us = $\_SESSION['username'];
124. **if** (empty($aname)) {
125. array\_push($errors, "Article name is required");
126. }
128. **if** (empty($content)) {
129. array\_push($errors, "Content is required");
130. }
132. **if**(count($errors) == 0) {
133. $sql = "INSERT INTO articles (aname, content, username, time) VALUES ('$aname', '$content', '$us', NOW());";
134. $result = $db->query($sql);
135. echo "<script> alert('Your article is successfully added.'); </script>";
136. }
137. }
139. **if** (isset($\_POST['change'])) {
140. $us = $\_SESSION['username'];
142. **if** (!empty($\_POST['uname'])) {
143. $uname = $\_POST['uname'];
144. $sql = "UPDATE user SET username = '$uname' WHERE username = '$us';"; // Не было '' и было ;
145. $result = $db->query($sql);
146. $\_SESSION['username'] = $uname;
147. }
149. $us = $\_SESSION['username'];
151. **if**(!empty($\_POST['fname'])) {
152. $fname = $\_POST['fname'];
153. $sql = "UPDATE user SET fname = '$fname' WHERE username = '$us';";
154. $result = $db->query($sql);
155. $\_SESSION['fullname'] = $fname;
156. }
158. **if** (!empty($\_POST['grade']) && (($\_POST['grade'] >= 1) && ($\_POST['grade'] <= 12))) {
159. $grade = (int) $\_POST['grade'];
160. $sql = "UPDATE user SET grade = '$grade' WHERE username = '$us';";
161. $result = $db->query($sql);
162. $\_SESSION['grade'] = $grade;
163. } **else** {
164. array\_push($errors, "Invalid data in grade field.");
165. }
167. **if** (!empty($\_POST['litera']) && (($\_POST['litera'] <= 'Z') && ($\_POST['litera'] >= 'A'))) {
168. $litera = $\_POST['litera'];
169. $sql = "UPDATE user SET litera = '$litera' WHERE username = '$us';";
170. $result = $db->query($sql);
171. $\_SESSION['litera'] = $litera;
172. }
174. **if** (!empty($\_POST['interests'])) {
175. $interests = $\_POST['interests'];
176. $sql = "UPDATE user SET interests = '$interests' WHERE username = '$us';";
177. $result = $db->query($sql);
178. $\_SESSION['interests'] = $interests;
179. }
181. header('location: profile.php');
182. }
184. **if** (isset($\_POST['delete'])) {
185. $us = $\_SESSION['username'];
186. // $conifirm = echo "<script> conifirm('Are you sure you want to delete this profile?'); </script>";
188. // if($conifirm) {
189. $sql = "DELETE FROM user WHERE username = '$us';";
190. $result = $db->query($sql);
192. session\_destroy();
193. unset($\_SESSION['username']);
194. header("location: login.php");
195. // }
196. }

9. Errors representing file (errors.php):

When errors are occurred, to represent them this code is called.

1. <?php **if** (count($errors) > 0): ?>
2. <div>
3. <?php **foreach** ($errors **as** $error): ?>
4. <p><?php echo $error; ?></p>
5. <?php **endforeach** ?>
6. </div>
7. <?php **endif** ?>

**TESTING:**

**Testing plan:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item | Valid | Invalid | Borderline | Success Criteria |
| Username field in registration. | Username starting with lowercase letter. | Username starting with number, uppercase letter or symbol. | Maximum username size – 15 symbols. | Data is saved in the database. |
| Email field in registration. | Email containing ‘@’ symbol, not ending with ‘@’ symbol and first symbol is lowercase letter. | String of characters. | Maximum email size – 30 symbols. | Data is saved in the database. |
| Grade field. | A number between 1 and 12. | A number less than 1 or more than 12. | Maximum value – 12, minimum – 1. | Data is saved in the database. |
| Letter field. | Uppercase English letter. | A symbol, number, lowercase letter or letter in another language. | Maximum size – 1. | Data is saved in the database. |

System testing:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No** | **Item to test** | **How I tested** | **Expected Result** | **Actual Result** | **Outcome** |
| 1 | “Your article” field | Wrote down Lorem ipsum into the field and pressed “Submit” button | Data will save to the database | Data didn’t save to the database | I added letter “t” into the “conent” word into the code. |
|  | | | | | |
| Retest 1 | Your article field. | Wrote down name of the article and Lorem ipsum | Data will save to the database | Data saved to the database | NA |
|  | | | | | |
| 2 | “Article’s name” field. | Wrote down “Math” into the field. | Data will be stored in the database. | Data stored into the database. | NA |
|  | | | | | |
| 3 | Registration form. | Wrote down information and pressed the button. | Data will be stored in the database and main page will appear on the screen. | Data didn’t store in the database and the error message occurred. | Deleted function in the code. |
|  | | | | | |
| Retest 3 | Registration form. | Wrote down information and pressed the button. | Data will be stored in the database and main page will appear on the screen. | Data stored in the database and main page appeared, but instead of the inputted password, word “password” appeared. | Added “$” sign before “password” in the code. |
|  | | | | | |
| Retest 3 | Registration form | Wrote down information and pressed the button. | Data will be stored in the database and main page will appear on the screen. | Data stored in the database and main page appeared. | NA. |
|  | | | | | |
| 4 | Articles’ posting | Enter main page. | All added articles will appear on the main page. | All added articles appeared on the main page. | NA. |
|  | | | | | |
| 5 | Login form. | Enter username and password. | Main page will appear on the screen. | Login form appeared again. | Changed the name of a table from “users” to “user”. |
|  | | | | | |
| Retest 5 | Login form. | Enter username and password. | Main page will appear on the screen. | Login for appeared again. | Deleted “$” in front of the “password”. |
|  | | | | | |
| Retest 5 | Login form | Enter username and password. | Main page will appear. | Main page appeared. | NA. |
|  | | | | | |
| 6 | Email confirm system testing. | Enter username, email and password. Go to the mailbox to confirm an email. | Message should be delivered to provided email. | Message didn’t deliver to provided email. | NA. |
|  | | | | | |
| 7 | Email confirm page. | Enter reference with email and confirm code. | “Registration finished. You may now log in.” message will appear. Value in the database of ccode will be 0 and confirmed will be 1. | “Registration finished. You may now log in.” message appeared. Value in the database of ccode is 0 and confirmed is 1. | NA. |
|  | | | | | |

Data validation testing table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No** | **Item to test** | **How I tested** | **Expected Result** | **Actual Result** | **Outcome** |
| **1** | Username field in registration. | Username starting with lowercase letter.  (valid) | “Registration finished. You may now log in.” message will appear. Username will appear in database. | “Registration finished. You may now log in.” message appeared. Username appeared in database. | NA. |
|  | | | | | |
| **2** | Username field in registration. | Username starting with number.  (invalid) | Error message will appear. | Error message appeared. | NA. |
|  | | | | | |
| **3** | Email field in registration. | Enter standard emal for gmail mailbox.  (valid) | “Registration finished. You may now log in.” message will appear. Email will appear in database. | “Registration finished. You may now log in.” message appeared. Email appeared in database. |  |
|  | | | | | |
| **4** | Email field in registration. | Enter email ending with ‘@’ symbol.  (invalid) | Error message will appear. | Error message appeared. | NA. |
|  | | | | | |
| **5** | Email field in registration. | Enter just a set of symbols.  (invalid) | Error message will appear. | Error message appeared. | NA. |
|  | | | | | |
| **6** | Email field in registration. | Enter just a set of symbols with ‘@’ in the middle.  (invalid) | Error message will appear. | “Registration finished. You may now log in.” message appeared. | NA. |
|  | | | | | |
| **7** | Grade field. | Enter number between 1 and 12.  (valid) | This number will appear on profile page. | This number appeared on profile page. | NA. |
|  | | | | | |
| **8** | Grade field. | Enter number over 12.  (invalid) | This number won’t appear on profile page. | This number did not appear on profile page. | NA. |
|  | | | | | |
| **9** | Grade field. | Enter number less than 1.  (invalid) | This number won’t appear on profile page. | This number did not appear on profile page. | NA. |
|  | | | | | |
| **10** | Letter field. | Enter upper case letter.  (valid) | This letter will appear on profile page. | This letter did appear on profile page. | NA. |
|  | | | | | |
| **11** | Letter field. | Enter text.  (invalid) | This text will not appear on profile page. | First entered letter appeared on the screen. | NA. |
|  | | | | | |
| **12** | Letter field. | Enter number.  (invalid) | This number will not appear on profile page. | This number did not appear on profile page. | NA. |
|  | | | | | |

System installation:

To install the social network I decide to use a parallel changeover without replacing old system, because old system works well and school community needs only addition in the form of social network. Both systems are going to work at the same time. The system does not need additional trainings, since the concept of social network is well-known and subconsciously easy to use.

User manual:

Form input objects:

|  |  |  |
| --- | --- | --- |
| **Object Name** | **Type** | **Use** |
| username | Input text | A form to type the username during registration and logging in. |
| email | Input text | A form to type user’s email during registration. |
| password\_1 | Input password | A form to type first password during registration. |
| password\_2 | Input password | A form to type second password to check if passwords match during registration. |
| password | Input password | A form to type the password during logging in |
| aname | Input text | A form to type the name of an article during article posting. |
| content | Input text | A form to type the content of an article during article posting. |
| uname | Input text | A form to type new username during profile changing. |
| fname | Input text | A form to type user’s full name during profile changing. |
| grade | Input text | A form to type the numerical |
| litera | Input text | A form to type the letter of the grade during profile changing. |
| interests | Input text | A form to type user’s interests during profile changing. |

Variables:

|  |  |  |  |
| --- | --- | --- | --- |
| **Object Name** | **Type** | **Page(s)** | **Use** |
| $username | String | server.php | A variable that carries username entered by user on reg.php and login.php pages. |
| $email | String | server.php | A variable that carries email address entered by user on reg.php page. |
| $password\_1 | String | server.php | A variable that carries password entered in the ‘password’ field on reg.php pages. |
| $password\_2 | String | server.php | A variable that carries password entered in the ‘confirm password’ field on reg.php page. |
| $ccode | Integer | server.php | A variable that carries confirm code that is randomly generated by php function. |
| $nosymlength | Integer | server.php | A variable that carries length of the $email variable withought ‘@’ symbol. |
| $dogpos | Integer | server.php | A variable that carries information about position of the ‘@’ symbol in the string ($email). |
| $i | Integer | server.php | A variable for iteration to work into the loop. |
| $sql | String | server.php | A variable that carries sql queries in the form of string. |
| $db | Function | server.php | A variable that provides connection to the database. |
| $result | String array | server.php | A variable that carries array of strings of the information that is selected from the database (all rows). |
| $data | String array | server.php | A variable that carries array of strings of the information that is selected from the database (only one row). |
| $message | String | server.php | A variable that carries message that is going to be send on email. |
| $errors | String array | server.php | A variable that carries array of errors that occur during different functions processing and then represented on pages. |
| $aname | String | server.php | A variable that carries the name of an article that is created by user. |
| $content | String | server.php | A variable that carries content in the form of text of the article that is written by user. |
| $us | String | server.php | A variable that carries username entered by user on warticle.php and profile\_change.php pages and during profile deleting process. |
| $uname | String | server.php | A variable that carries new username of the user on profile\_change.php page. |
| $fname | String | server.php | A variable that carries full name of the user on profile\_change.php page. |
| $grade | Integer | server.php | A variable that carries class number of the user on profile\_change.php page. |
| $litera | String | server.php | A variable that carries grade’s letter of the user on profile\_change.php page. |
| $interests | String | server.php | A variable that carries information about interests of the user on profile\_change.php page. |
| $\_SESSION | Array | server.php | A variable that carries username that can be used on any page. |
| $\_POST | Array | server.php | A variable that provides information from form |

Evaluation:

**1st objective:** “to create a universal navigation bar for every web page with home, articles and logout sections.”

I have mostly achieved this objective, because I was competent enough in HTML and CSS. Navigation bar contains “Home”, “Article”, “Profile” and “Logout” sections and is placed on every webpage except login and registration. The only con of my navigation bar is that it is hard to understand where “Article” reference leads, because after pressing this link user is sent to article creation page. In further working I may change the name of this section or give an instruction for user about its’ functionality.

**2nd objective:** “to create a sidebar for homepage with profile, advice of the day and last blog sections.”

I absolutely have not achieved this objective, since some parts of the sidebar were added in other sections and there is no further need in it. For profile I created another web page, where users can logout, change and delete profile. Advices were added on the home page, where just a list of advice is represented. Last blog was not included into my project, since articles are represented from newest to latest, so the first article is the last article that was added. In my opinion I’ve done even more than was claimed in this objective, because sidebar is very small and there is not enough space for these sections. The bad thing about such alternative is that articles are sorted by the time they were written in future project I may add the button to change sorting criteria.

**3rd objective:** “to draw a logo for social network.”

I achieved this section by painting logo in the Photoshop. The logo in the footer looks bad, because it is underlined with white color.

**4th objective:** “to create working registration and login forms.”

I mostly achieved this objective. I have working registration with username, email, password and confirm password fields. My code checks if inputted username or email exists in database, if passwords match and makes validation of this data. Validation is poor. My program checks for password and username length, position of ‘@’ symbol in the inputted email and starting character of username string. My program informs user about errors only after he or she presses the button and all data in all fields updates, so user should start entering the data from the beginning. To avoid it in the future I may add Ajax on the registration page. Login form works well. My program checks for username and password in the database and allows user to enter via his or her personal data. All forms are not responsive. In the future I may add more CSS to make it more responsive.