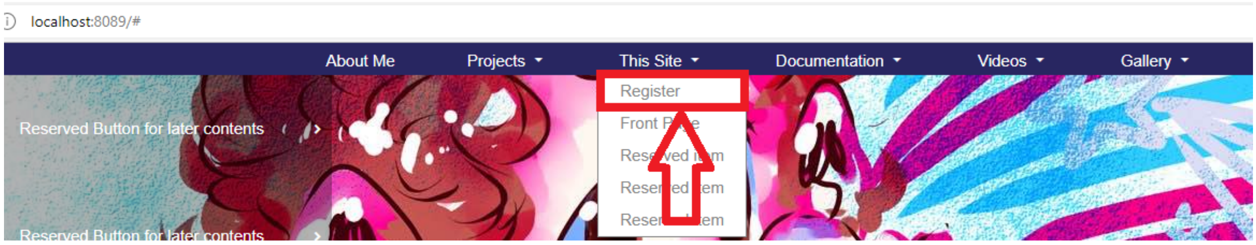
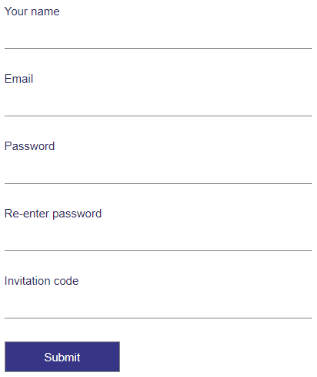
First, start up the mini-server, and then go to the localhost test page, note that for the mini-server, the default port for HTTP is 8089.

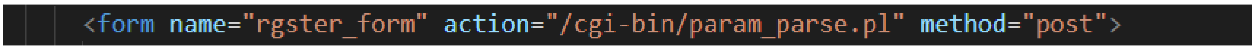


Go to the navigation bar and choose “This Site” -> “Register”, you will be guided to the registration page, which is specifically designed for this task.

On the registration page, there is a form for gathering users’ information.

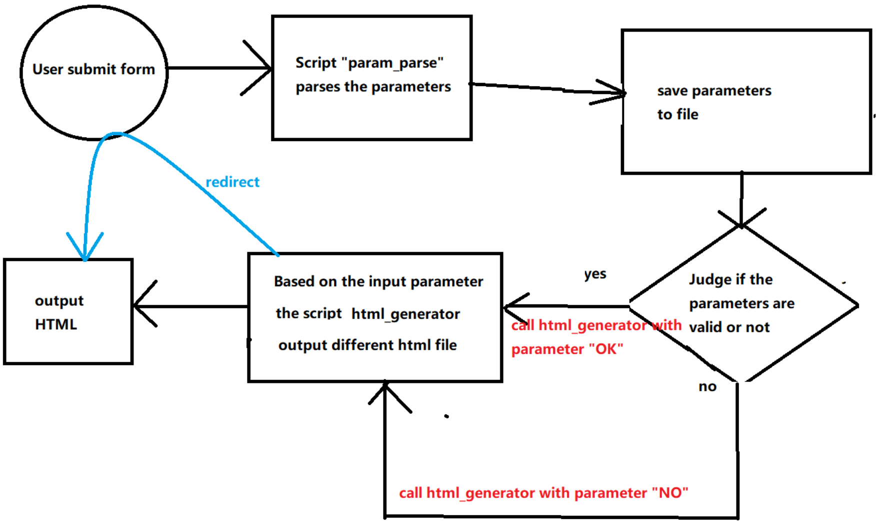


In the HTML file, the registration page has a form structure.

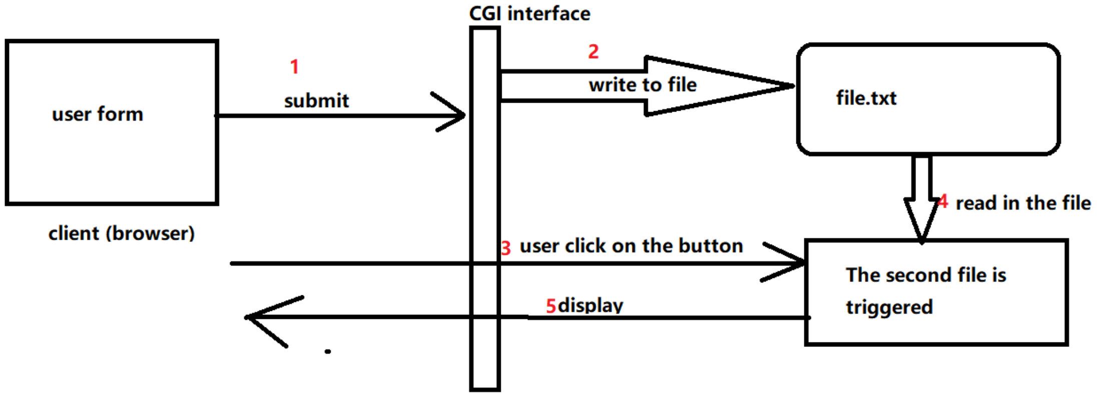


```
<form name="rgster_form" action="/cgi-bin/param_parse.pl" method="post">
```

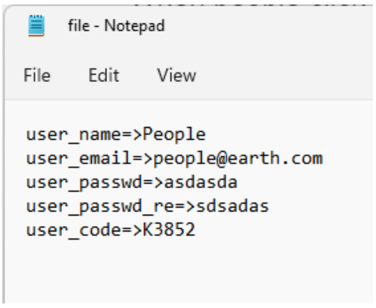
The registration page I designed has the logic shown in the figure below.


User submit form
Script "param_parse" parses the parameters
save parameters to file
Judge if the parameters are valid or not
yes
call html_generator with parameter "OK"
no
call html_generator with parameter "NO"
Based on the input parameter the script html_generator output different html file
output HTML
redirect

After receiving the form submitted by the user, the script “param_parse” will parse the parameter and save the parameter to a file. Then, it will judge if the parameter is valid or not, and based on the judgment invoke another Perl script html_generator to generate HTML differently.

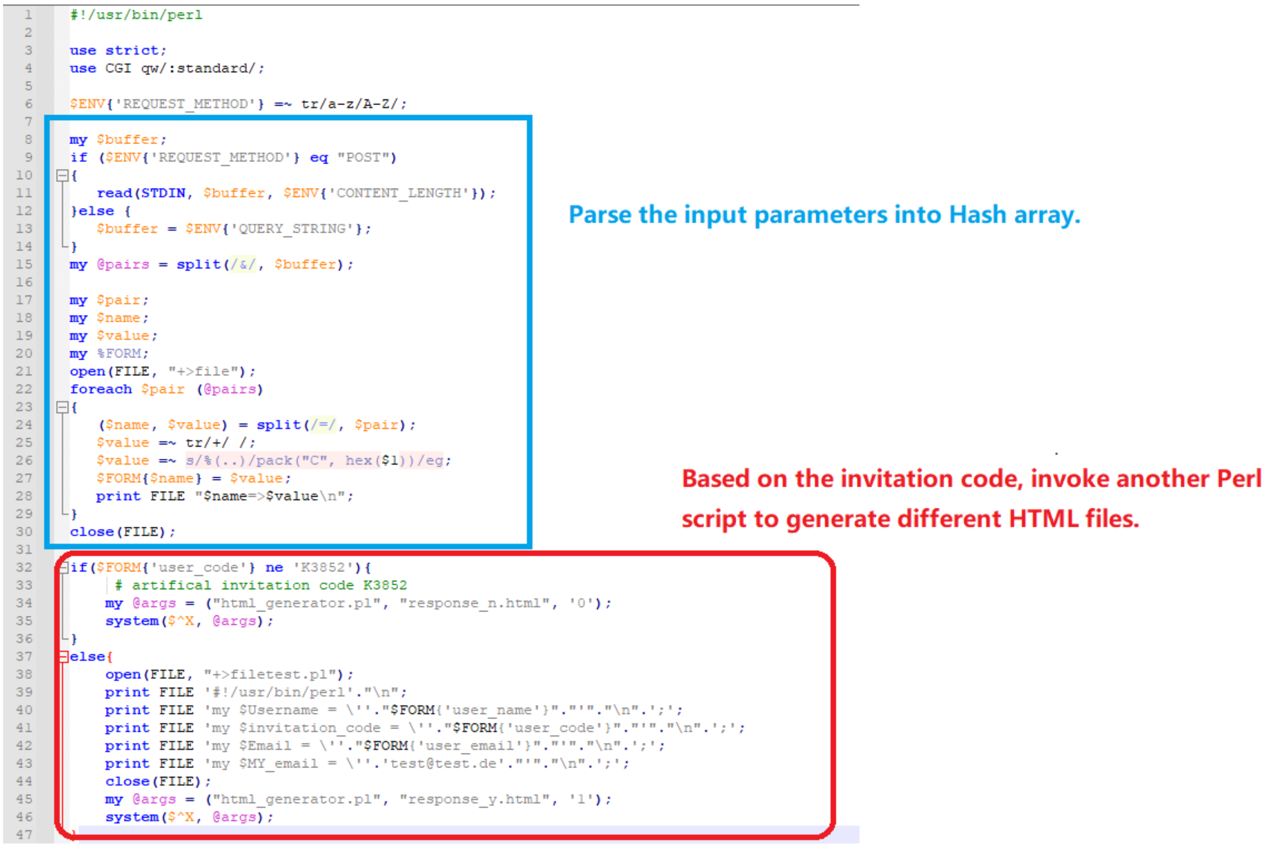


When people click on the button to submit the form, the form will be POSTED to the CGI script in the server, the script that is responsible for parsing and writing the parameters is param_parse.pl. The parsing result is shown below.



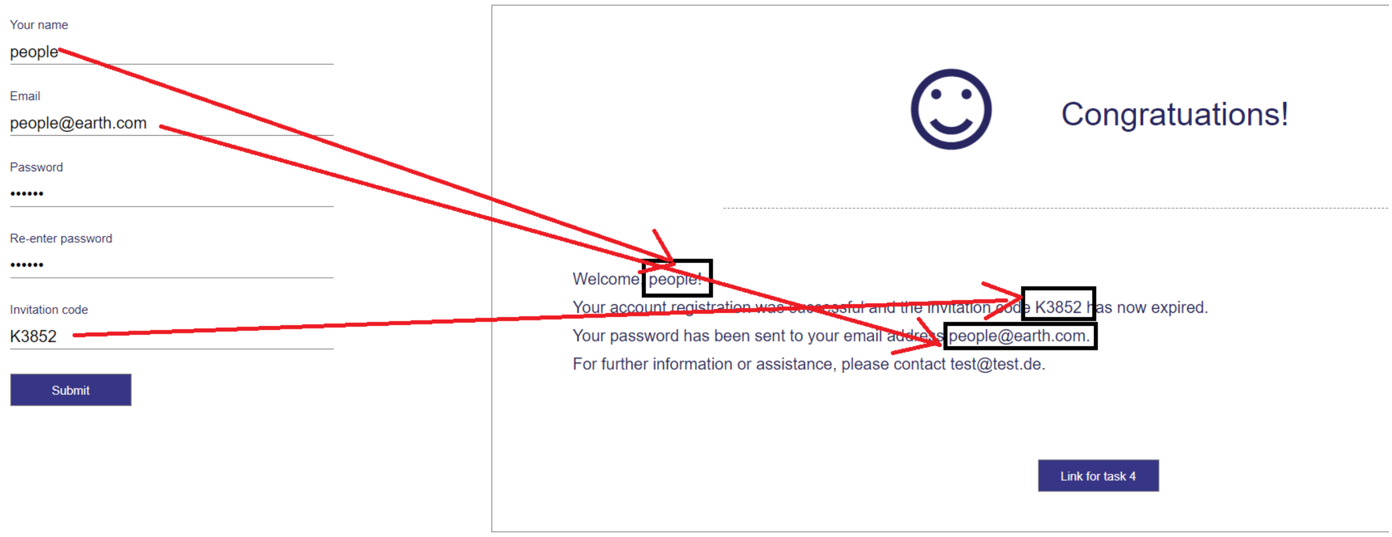
```
user_name=>People
user_email=>people@earth.com
user_passwd=>asdasda
user_passwd_re=>sdsadas
user_code=>K3852
```

And the code of param_parse.pl is shown below.


Parse the input parameters into Hash array.
Based on the invitation code, invoke another Perl script to generate different HTML files.

```perl
#!/usr/bin/perl

use strict;
use CGI qw/:standard/;

$ENV{'REQUEST_METHOD'} =~ tr/a-z/A-Z/;

my $buffer;
if ($ENV{'REQUEST_METHOD'} eq "POST")
{
    read(STDIN, $buffer, $ENV{'CONTENT_LENGTH'});
}else {
    $buffer = $ENV{'QUERY_STRING'};
}
my @pairs = split(/&/, $buffer);

my $pair;
my $name;
my $value;
my %FORM;
open(FILE, "+>file");
foreach $pair (@pairs)
{
    ($name, $value) = split(/=/, $pair);
    $value =~ tr/+/ /;
    $value =~ s/%(..)/pack("C", hex($1))/eg;
    $FORM{$name} = $value;
    print FILE "$name=>$value\n";
}
close(FILE);

if($FORM{'user_code'} ne 'K3852'){
     # artifical invitation code K3852
    my @args = ("html_generator.pl", "response_n.html", '0');
    system($^X, @args);
}
else{
    open(FILE, "+>filetest.pl");
    print FILE '#!/usr/bin/perl'."\n";
    print FILE 'my $Username = \''."$FORM{'user_name'}"."'"."\n".';';
    print FILE 'my $invitation_code = \''."$FORM{'user_code'}"."'"."\n".';';
    print FILE 'my $Email = \''."$FORM{'user_email'}"."'"."\n".';';
    print FILE 'my $MY_email = \''.'test@test.de'."'"."\n".';';
    close(FILE);
    my @args = ("html_generator.pl", "response_y.html", '1');
    system($^X, @args);
}
```

Then a dynamic website is generated by HTML generator



Your name
people

Email
people@earth.com

Password
••••••

Re-enter password
••••••

Invitation code
K3852

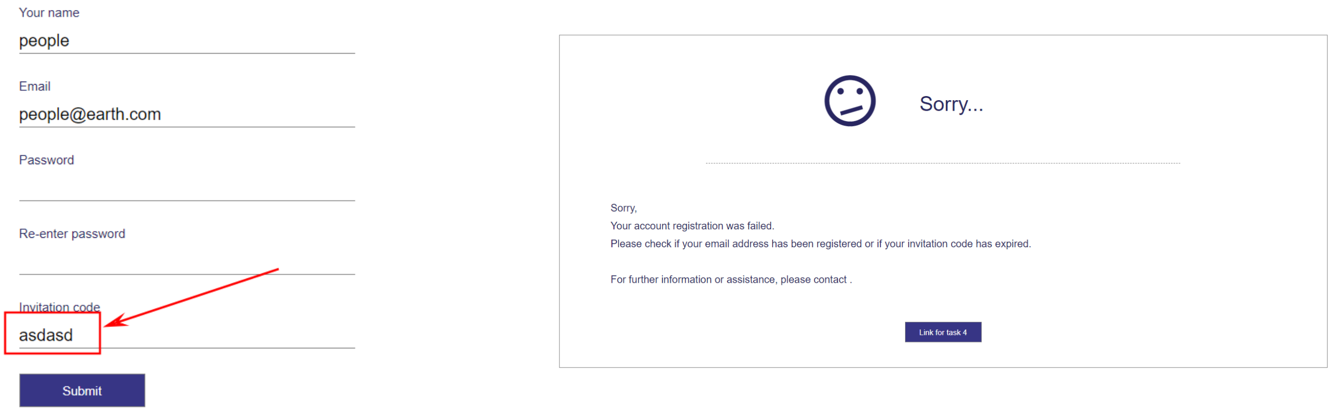
Submit

Congratuations!

Welcome people!
Your account registration was successful and the invitation code K3852 has now expired.
Your password has been sent to your email address people@earth.com.
For further information or assistance, please contact test@test.de.

Link for task 4

If one enters the invitation code that is different from K3852, a fail page will be generated.



Your name
people

Email
people@earth.com

Password

Re-enter password

Invitation code
asdasd

Submit

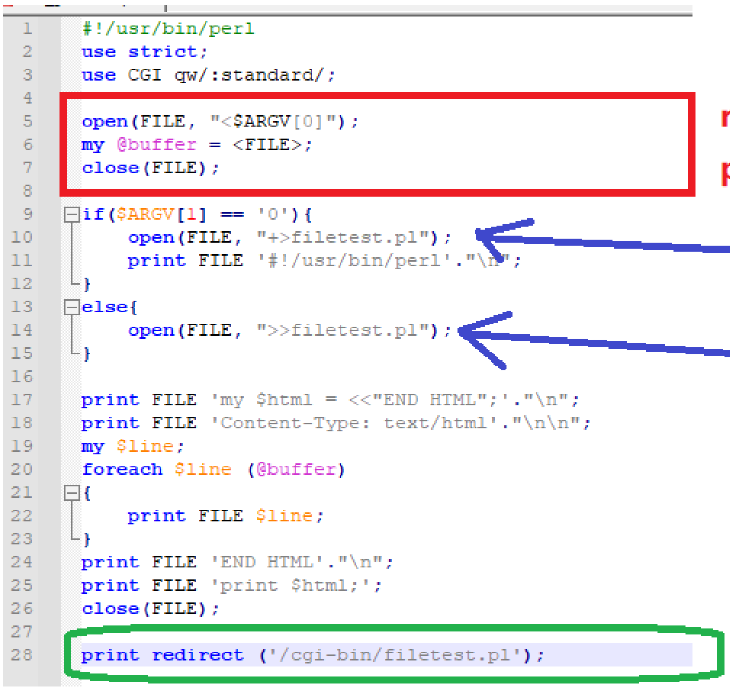
Sorry...

Sorry,
Your account registration was failed.
Please check if your email address has been registered or if your invitation code has expired.

For further information or assistance, please contact .

Link for task 4

The code of html_generator is shown below



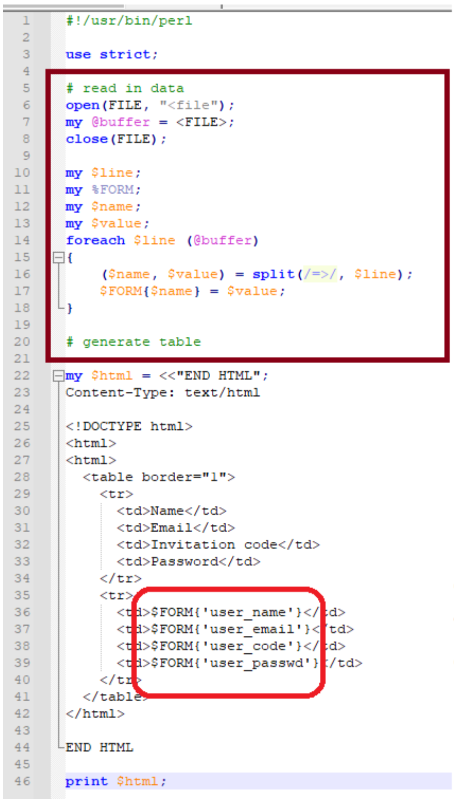
```
#!/usr/bin/perl
use strict;
use CGI qw/:standard/;

open(FILE, "<$ARGV[0]");
my @buffer = <FILE>;
close(FILE);

if($ARGV[1] == '0'){
    open(FILE, "+>filetest.pl");
    print FILE '#!/usr/bin/perl'."\n";
}
else{
    open(FILE, ">>filetest.pl");
}

print FILE 'my $html = <<"END HTML";'."\n";
print FILE 'Content-Type: text/html'."\n\n";
my $line;
foreach $line (@buffer)
{
    print FILE $line;
}
print FILE 'END HTML'."\n";
print FILE 'print $html;';
close(FILE);

print redirect ('/cgi-bin/filetest.pl');
```

read in the HTML file based on the input parameter

create HTML file

redirect to the generated page

For creating a page containing a table dynamically displaying the data, it is finished by the Perl script “for_task.pl”.

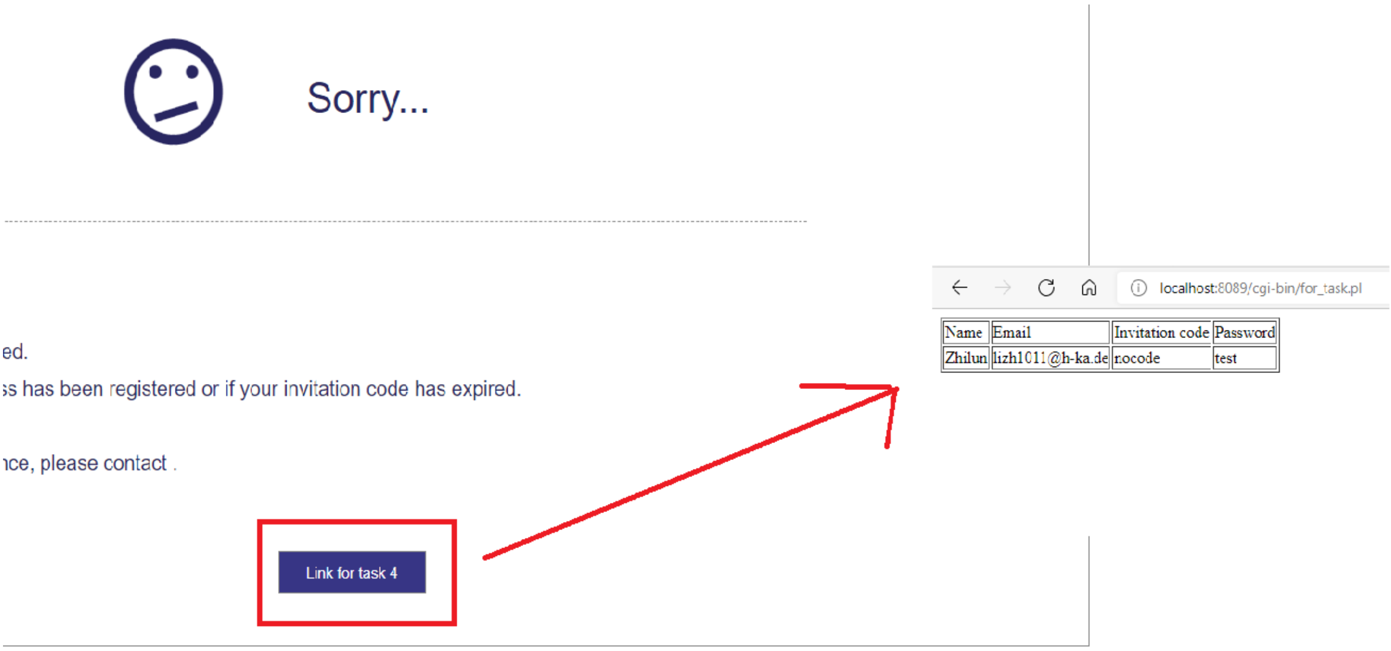


```perl
#!/usr/bin/perl

use strict;

# read in data
open(FILE, "<file");
my @buffer = <FILE>;
close(FILE);

my $line;
my %FORM;
my $name;
my $value;
foreach $line (@buffer)
{
    ($name, $value) = split(/=>/, $line);
    $FORM{$name} = $value;
}

# generate table

my $html = <<"END HTML";
Content-Type: text/html

<!DOCTYPE html>
<html>
<html>
  <table border="1">
    <tr>
      <td>Name</td>
      <td>Email</td>
      <td>Invitation code</td>
      <td>Password</td>
    </tr>
    <tr>
      <td>$FORM{'user_name'}</td>
      <td>$FORM{'user_email'}</td>
      <td>$FORM{'user_code'}</td>
      <td>$FORM{'user_passwd'}</td>
    </tr>
  </table>
</html>

END HTML

print $html;
```

Read in data and save them in to Hash array.

call the Hash array to dynamically create the page

When people click on the button, a dynamically generated table will be displayed, on matter if the invitation code is correct or not.



Sorry...

ed.

ıs has been registered or if your invitation code has expired.

ıce, please contact .

Name	Email	Invitation code	Password
Zhilun	lizh1011@h-ka.de	nocode	test