Dynamic Web Design

Alpha prototype Report（ Group 1 ）

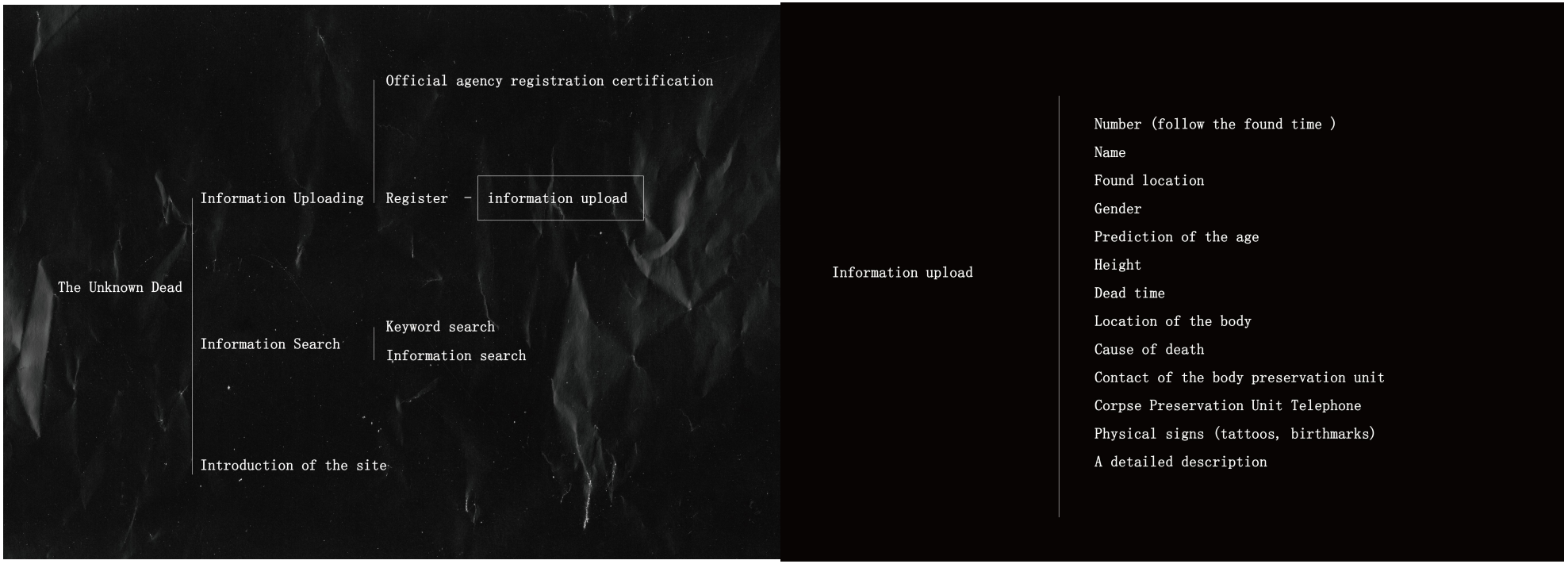
Group Member: Ziyu Du; Luke Cao; Luoying Bao

https://ziyudu.edinburgh.domains/fatfree/DeadPeopleSystem/index

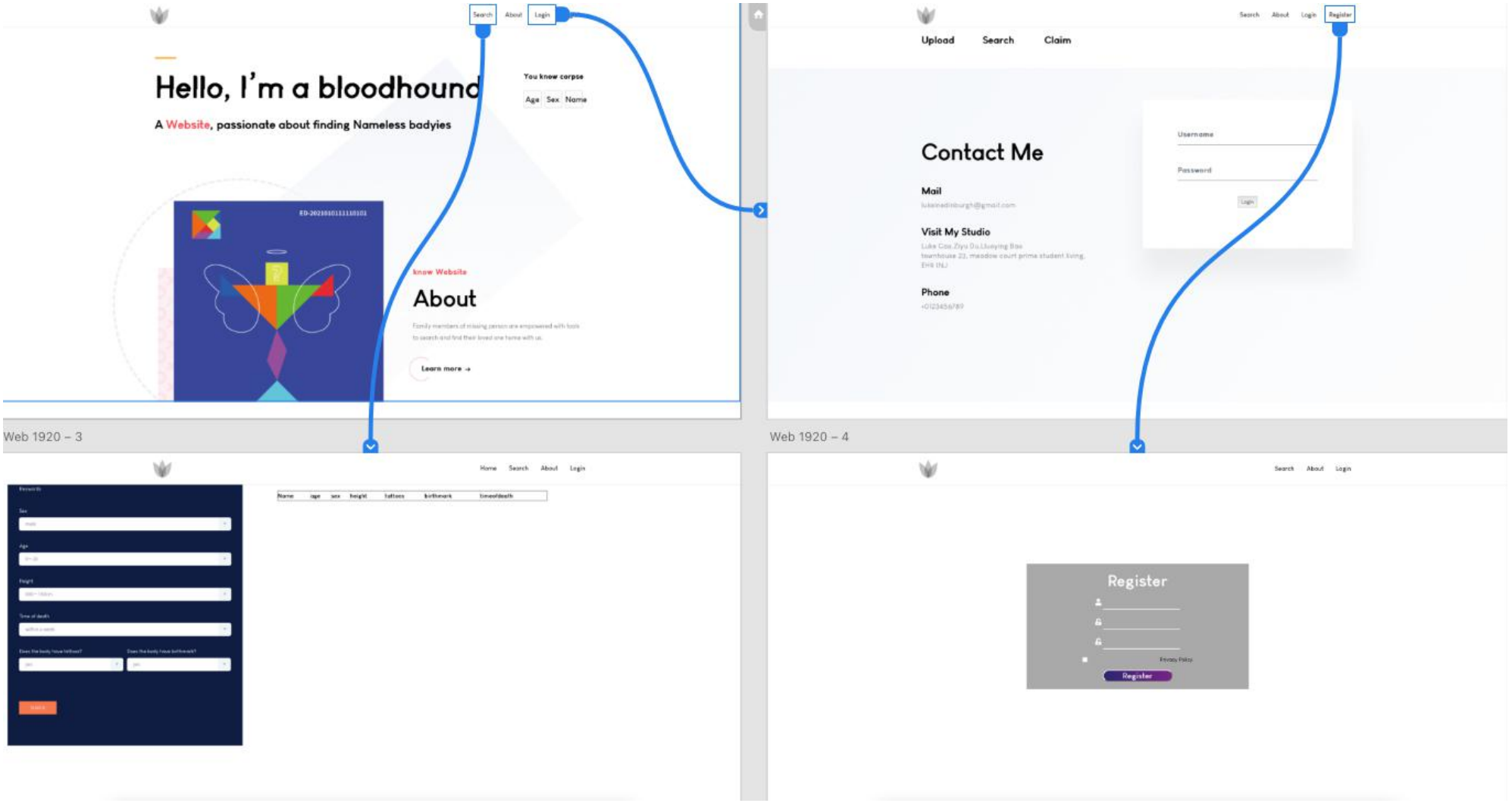
Background and Motivation

Expensive funeral costs are a serious problem and appear to have worsened over the years. According to these expensive funeral costs and other special circumstances, during the popularity of Covid-19, medical examiners' offices and local news has indicated that the number of unidentified bodies is climbing. So in our project, we are aiming to establish a website to connect police departments in different areas and help people search and claim unidentified bodies by posting body information on.

Introduction



1.0 User flow[1]



2.0 User flow[1]

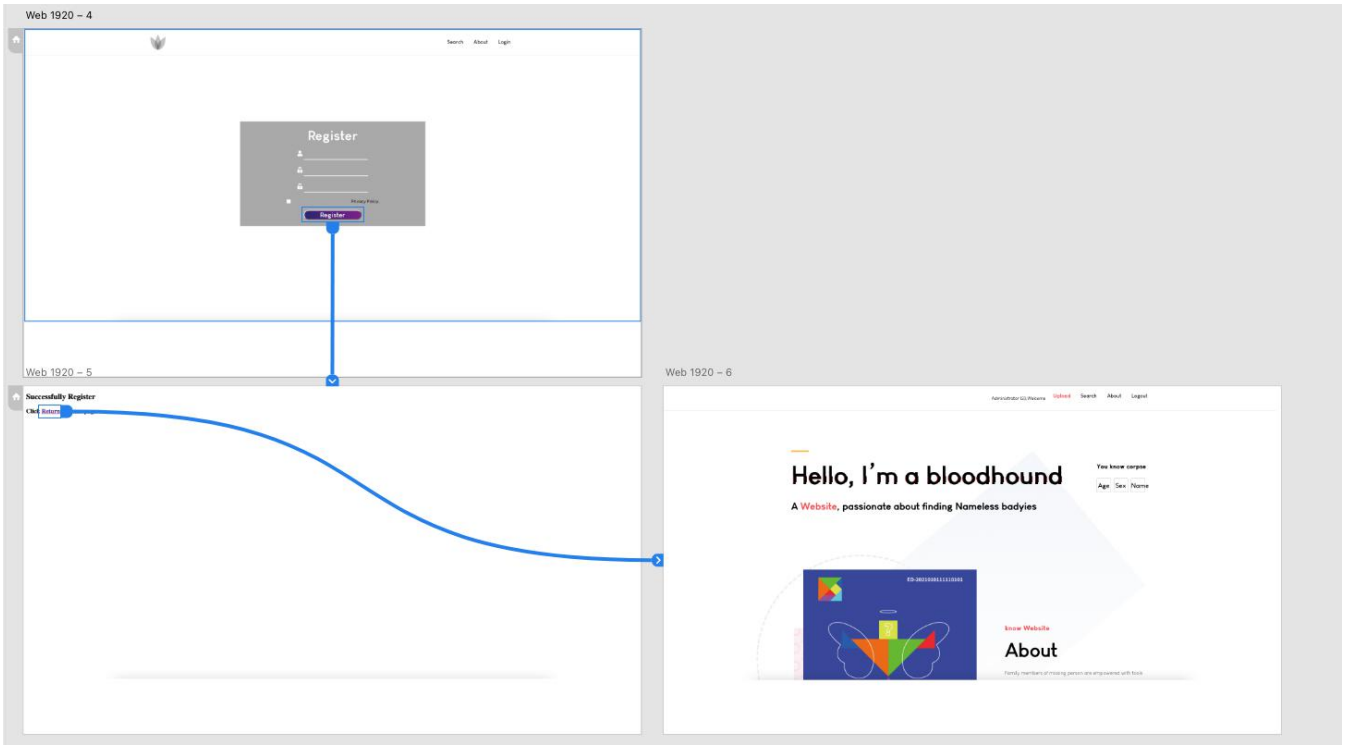


Figure: flow chart [2]

1.1 Front End

The Alpha prototype of our project contains three core functions: search, administrator register and login. These functions are mainly composed of html, which converts part of our ui design into simple lines of code. At the same time, we used some things on bootstrap to complete our front-end design, such as the animation design of the search bar and homepage.

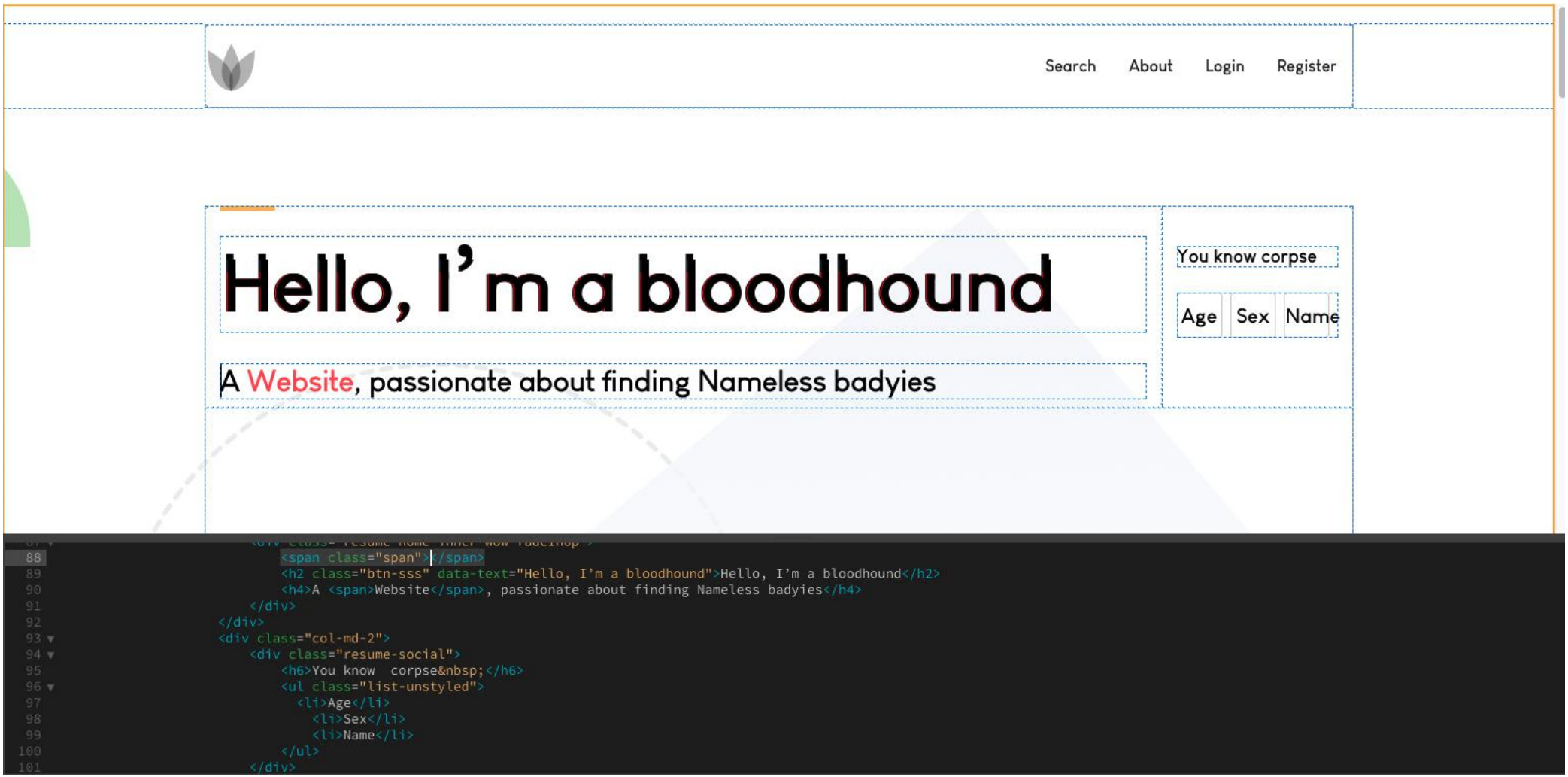
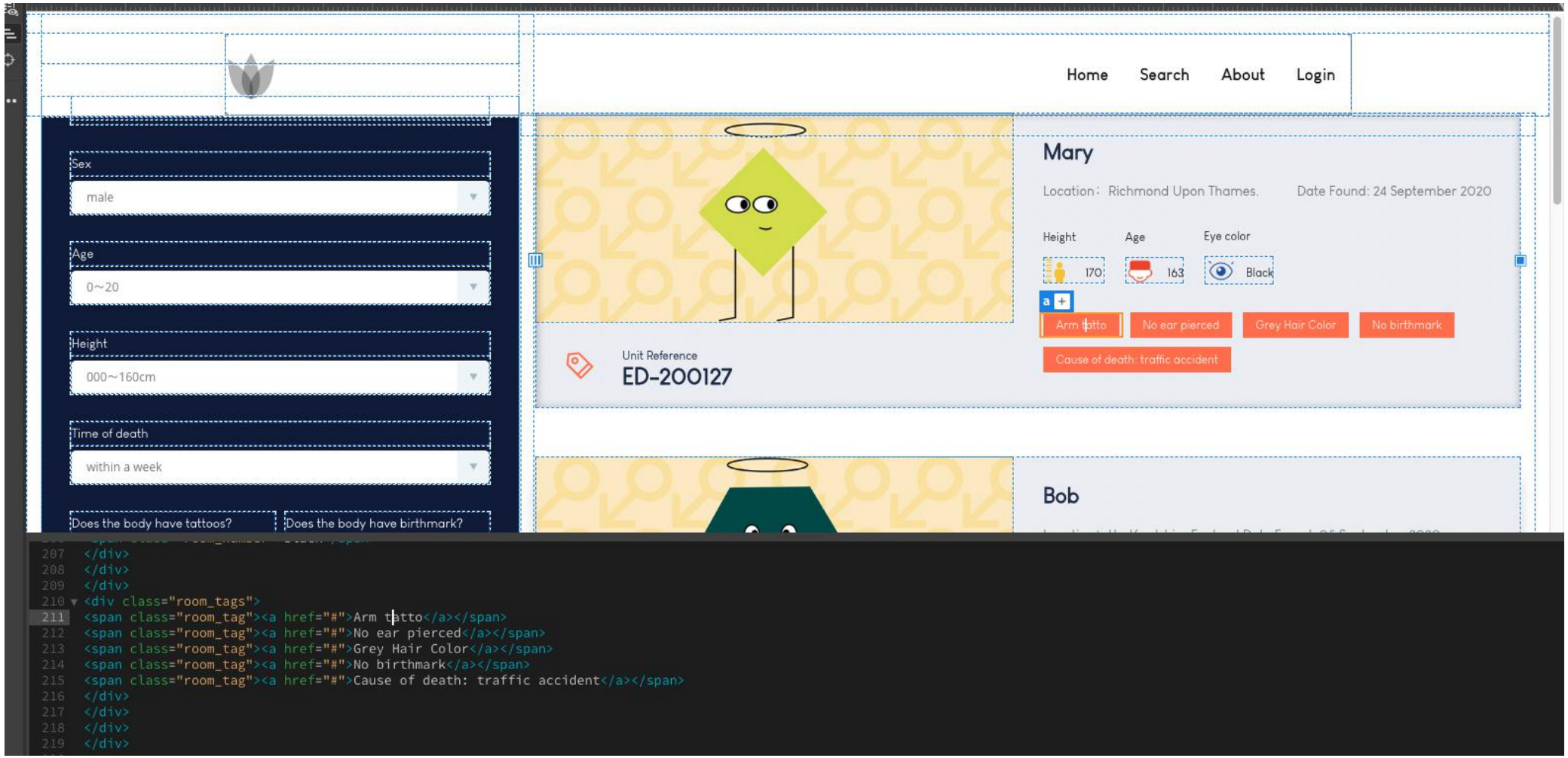


figure: homepage [2]




```
</div>
</div>
</div>
<div class="room_tags">
<span class="room_tag"><a href="#">Arm tatto</a></span>
<span class="room_tag"><a href="#">No ear pierced</a></span>
<span class="room_tag"><a href="#">Grey Hair Color</a></span>
<span class="room_tag"><a href="#">No birthmark</a></span>
<span class="room_tag"><a href="#">Cause of death: traffic accident</a></span>
</div>
</div>
</div>
</div>
```

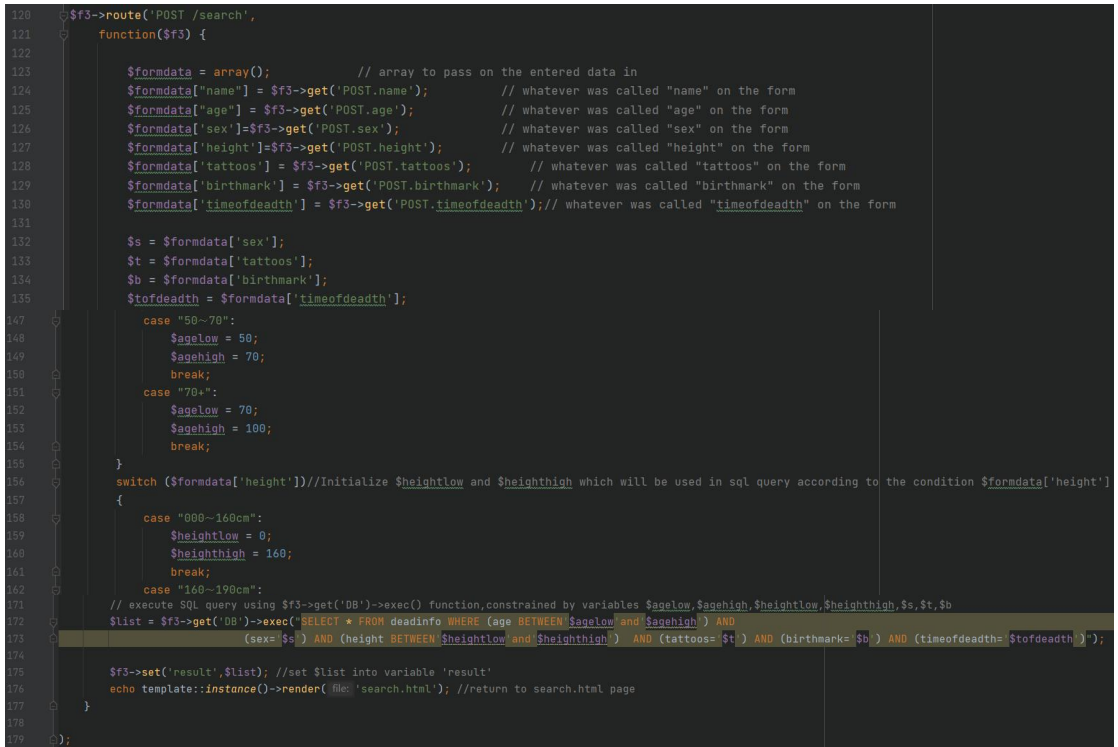
Figure: search page [2]

1.2 Back End

The Alpha prototype of our project contains three core functions: search, administrator register and login. These three functions are all completed by f3-fatfree-framework in index.php under the path of DeadPeopleSystem on server. The methodologies of these three functions are quite similar, information on form from front-end is loaded into bac- end using http POST request and information will be processed such as insert, delete, query and modify. Then, the results of process from back-end will be returned to html page.

Search

The code of search function is as followed:



```
$f3->route('POST /search',
    function($f3) {

        $formdata = array();           // array to pass on the entered data in
        $formdata["name"] = $f3->get('POST.name');          // whatever was called "name" on the form
        $formdata["age"] = $f3->get('POST.age');            // whatever was called "age" on the form
        $formdata['sex']=$f3->get('POST.sex');              // whatever was called "sex" on the form
        $formdata['height']=$f3->get('POST.height');        // whatever was called "height" on the form
        $formdata['tattoos'] = $f3->get('POST.tattoos');        // whatever was called "tattoos" on the form
        $formdata['birthmark'] = $f3->get('POST.birthmark');    // whatever was called "birthmark" on the form
        $formdata['timeofdeadth'] = $f3->get('POST.timeofdeadth');// whatever was called "timeofdeadth" on the form

        $s = $formdata['sex'];
        $t = $formdata['tattoos'];
        $b = $formdata['birthmark'];
        $tofdeadth = $formdata['timeofdeadth'];
          case "50~70":
              $agelow = 50;
              $agehigh = 70;
              break;
          case "70+":
              $agelow = 70;
              $agehigh = 100;
              break;
      }
       switch ($formdata['height'])//Initialize $heightlow and $heighthigh which will be used in sql query according to the condition $formdata['height']
       {
          case "000~160cm":
              $heightlow = 0;
              $heighthigh = 160;
              break;
          case "160~190cm":
      // execute SQL query using $f3->get('DB')->exec() function,constrained by variables $agelow,$agehigh,$heightlow,$heighthigh,$s,$t,$b
       $list = $f3->get('DB')->exec("SELECT * FROM deadinfo WHERE (age BETWEEN '$agelow' and '$agehigh') AND
                          (sex='$s') AND (height BETWEEN '$heightlow' and '$heighthigh')  AND (tattoos='$t') AND (birthmark='$b') AND (timeofdeadth='$tofdeadth')");

       $f3->set('result',$list); //set $list into variable 'result'
       echo template::instance()->render( file: 'search.html'); //return to search.html page
   }
);
```

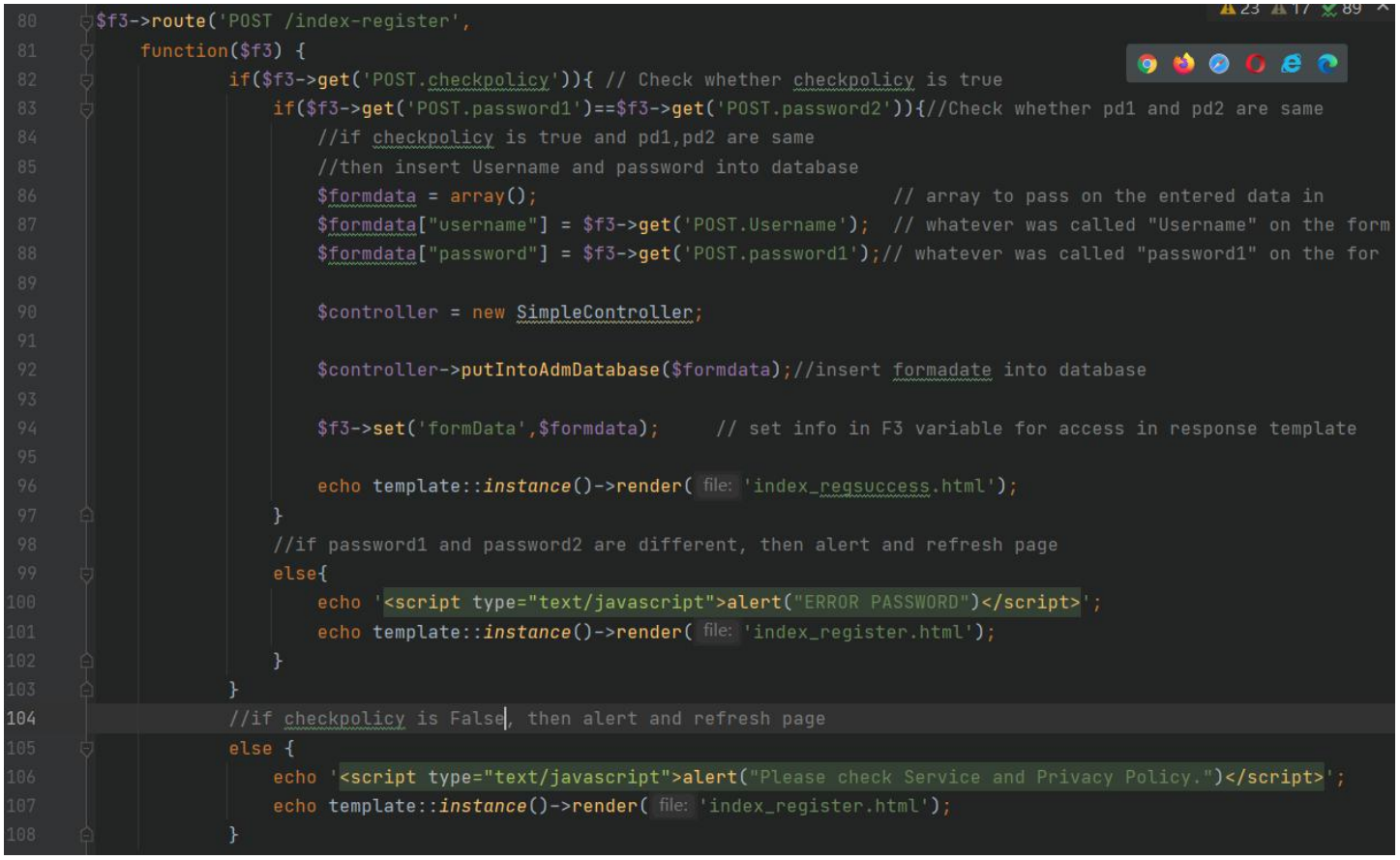
Figure: search function [1]

The php codes of search function can be divided into four sections. The first section from line 123 to 135 can be denoted as loading data. In this section, whenever the selections on form(front end) are called, some related data will be stored into array called formdata. The second section aims at initializing eight variables which will be used in SQL query. The SQL query can be considered as the

third section of search function which is the core algorithm for overall function. In this section the SQL

query will be executed by applying exec() function and the this SQL query will be constrained by eight variables initialized in section 2. Finally, the query results will be returned to front end and refresh page.

Administrator register (The code of administrator register function is as followed)



```
$f3->route('POST /index-register',
    function($f3) {
            if($f3->get('POST.checkpolicy')){ // Check whether checkpolicy is true
                if($f3->get('POST.password1')==$f3->get('POST.password2')){//Check whether pd1 and pd2 are same
                    //if checkpolicy is true and pd1,pd2 are same
                    //then insert Username and password into database
                    $formdata = array();                                // array to pass on the entered data in
                    $formdata["username"] = $f3->get('POST.Username');  // whatever was called "Username" on the form
                    $formdata["password"] = $f3->get('POST.password1');// whatever was called "password1" on the for

                    $controller = new SimpleController;

                    $controller->putIntoAdmDatabase($formdata);//insert formadate into database

                    $f3->set('formData',$formdata);     // set info in F3 variable for access in response template

                    echo template::instance()->render( file: 'index_regsuccess.html');
                }
                //if password1 and password2 are different, then alert and refresh page
                else{
                    echo '<script type="text/javascript">alert("ERROR PASSWORD")</script>';
                    echo template::instance()->render( file: 'index_register.html');
                }
            }
            //if checkpolicy is False, then alert and refresh page
            else {
                echo '<script type="text/javascript">alert("Please check Service and Privacy Policy.")</script>';
                echo template::instance()->render( file: 'index_register.html');
            }
```

Figure: php code of administrator register [1]

This function aims at solving the problem of administrator registering. There are two constrains if an administrator registers successfully. Registrant must agree to the privacy policy and the confirmed password mush be same as the one they typed in at the beginning. If there is a condition that is not met, alert interface will pop up and the page will be refreshed.

```php
public function putIntoAdmDatabase($data) {
    $this->loginMapper->username = $data["username"];                    // set value for "username" field
    $this->loginMapper->password = $data["password"];                 // set value for "password" field
    $this->loginMapper->save();                                     // save new record with these fields
}
```

Figure: putIntoAdmDatabase function [1]

Administrator login (The code of administrator login function is as followed)

```php
$f3->route('POST /index',
    function($f3) {
        $controller = new SimpleController;
        if ($controller->loginUser($f3->get('POST.Username'), $f3->get('POST.Password'))) { // user is recognised
            $f3->set('SESSION.userName', $f3->get('POST.Username'));
            // note that this is a global that will be available elsewhere
            header( header: "location:/fatfree/DeadPeopleSystem/index-user");
            // will always go to index-user after successful login
        }
        else{
            echo "<script type='text/javascript'>alert('ERROR password or username')</script>";
            // return to login page with the message that there was an error in the credentials
            echo Template::instance()->render( file: 'index.html');
        }
    }
);
```

Figure: php code of administrator login [1]

This function waits for POST request in index page, loginUser function is used to verify the correctness of Username and Password. Whenever Username and Password is true, we will always go to index-user after successful login.

What to do next?

1. We also need to make an upload page and some simple information introduction page. The upload page includes the upload of the photo information of the deceased, as well as the map marking function, and users can save the location of the unnamed corpse online.

2. Our user interface design is based on the bootstrap template. We would have thought that this would reduce the workload but it would cause more troubles. We will redesign the user interface and the front-end design part but the back-end functions will remain constant.

3. In the back-end design, we also have many parts that need to be improved; for example, after retrieving the information of an unnamed corpse, the information display of the corpse currently only displays text and table information, and cannot view a series of information such as pictures.

4. We want to try to use three.js to use 3D interactive features on the homepage of our website. This may be difficult but we will definitely try.

Reference

[1] FROM TUTORIAL:

```
public function loginUser($user, $pwd) {        // very simple login -- no use of encryption, hashing etc.
    $auth = new \Auth($this->loginMapper, array('id'=>'username', 'pw'=>'password'));   // fields in table
    return $auth->login($user, $pwd);           // returns true on successful login
}
```

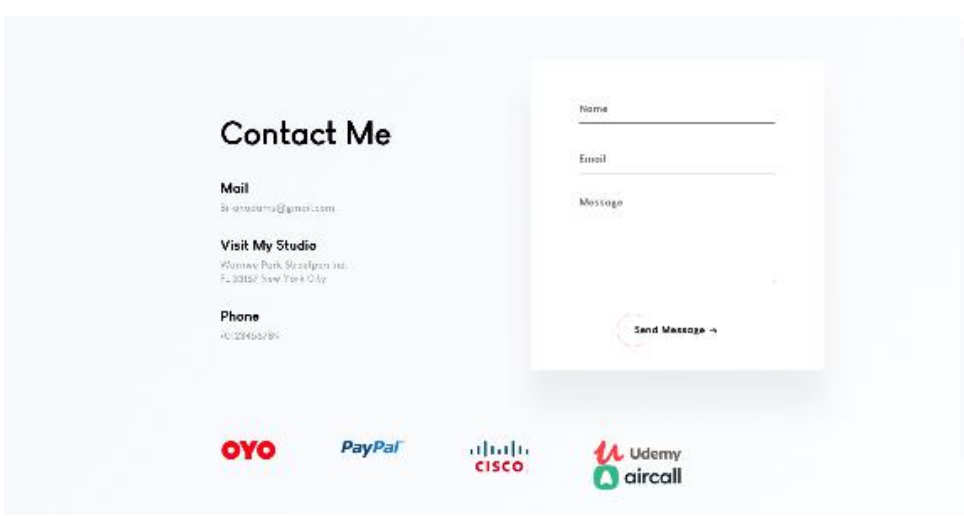
```
 public function getData() {
     $list = $this->deadinfo->find();
     return $list;
 }
```

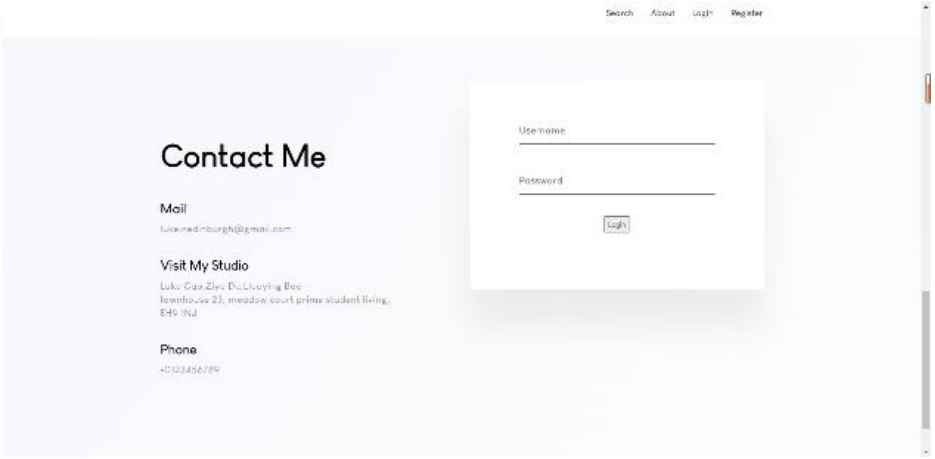
```
public function putIntoinfoDatabase($data) {
    $this->deadinfo->name = $data["name"];                  // set value for "name" field
    $this->deadinfo->age = $data["colour"];                // set value for "colour" field
    $this->deadinfo->save();                                    // save new record with these fields
}
```

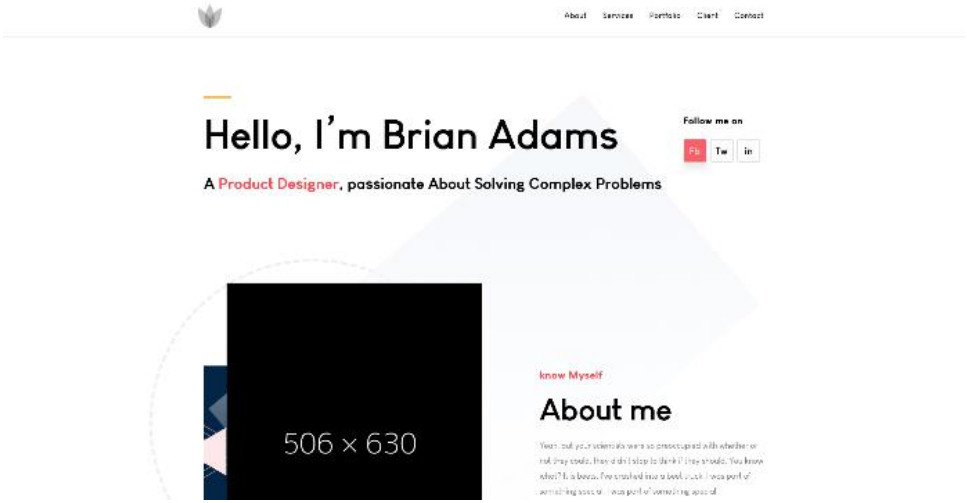
[2] front end Template:

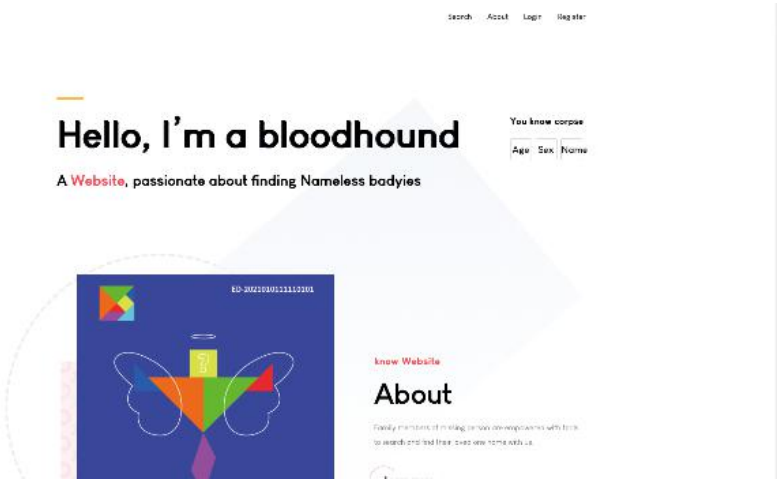
https://ui8.net/luova-studio/products/sleek-portfolio-html-template

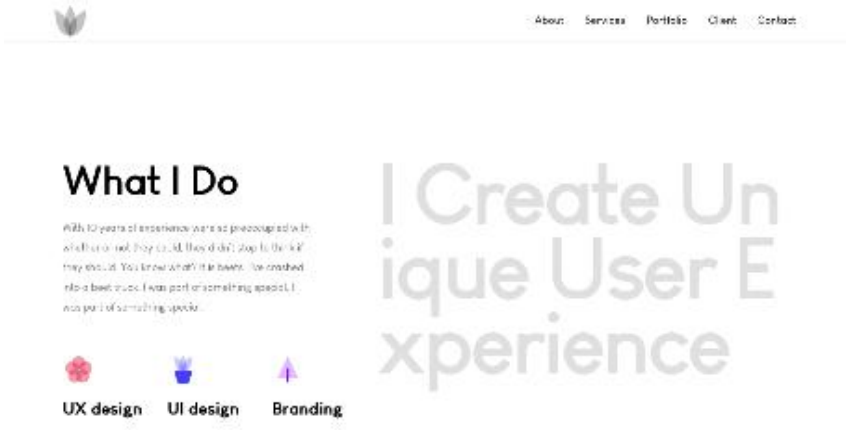
Template **VS** **Our Works**







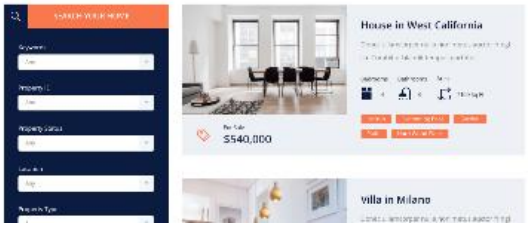




[3] front end Template：

https://www.8m2.net/2019%e5%b9%b4%e5%a4%a7%e6%b0%94%e6%b5%b7%e8%93%9d%e8%89%b2%e7%81%b0%e8%89%b2%e6%88%bf%e5%9c%b0%e4%ba%a7%e7%bd%91%e7%ab%99%e6%a8%a1%e6%9d%bf/

Template **VS** **Our works**



Division of labor

1. The theme of the webpage: Ziyu Du; Luke Cao; Luoying Bao
2. Main functions of the webpage & information structure: Ziyu Du; Luke Cao; Luoying Bao
3. Visual design: Luoying Bao

2. Front end design：Ziyu Du; Luke Cao;

3. Back-end design:Ziyu Du;