



# **WEB PROGRAMIRANJE PHP2**

## **DOKUMENTACIJA PROJEKTA**

Student:

Ivan Laba 168/15

Beograd 2018.

## SADRŽAJ

1.Opis funkcionalnosti .....	1
2. Skica struktura stranica .....	1
3. Dijagram baze podataka .....	4
3.MVC Organizacija .....	5
Kontroleri .....	5
Modeli .....	6
View .....	8
4. javascript kod.....	8
5. PHP kod.....	9

## 1.Opis funkcionalnosti

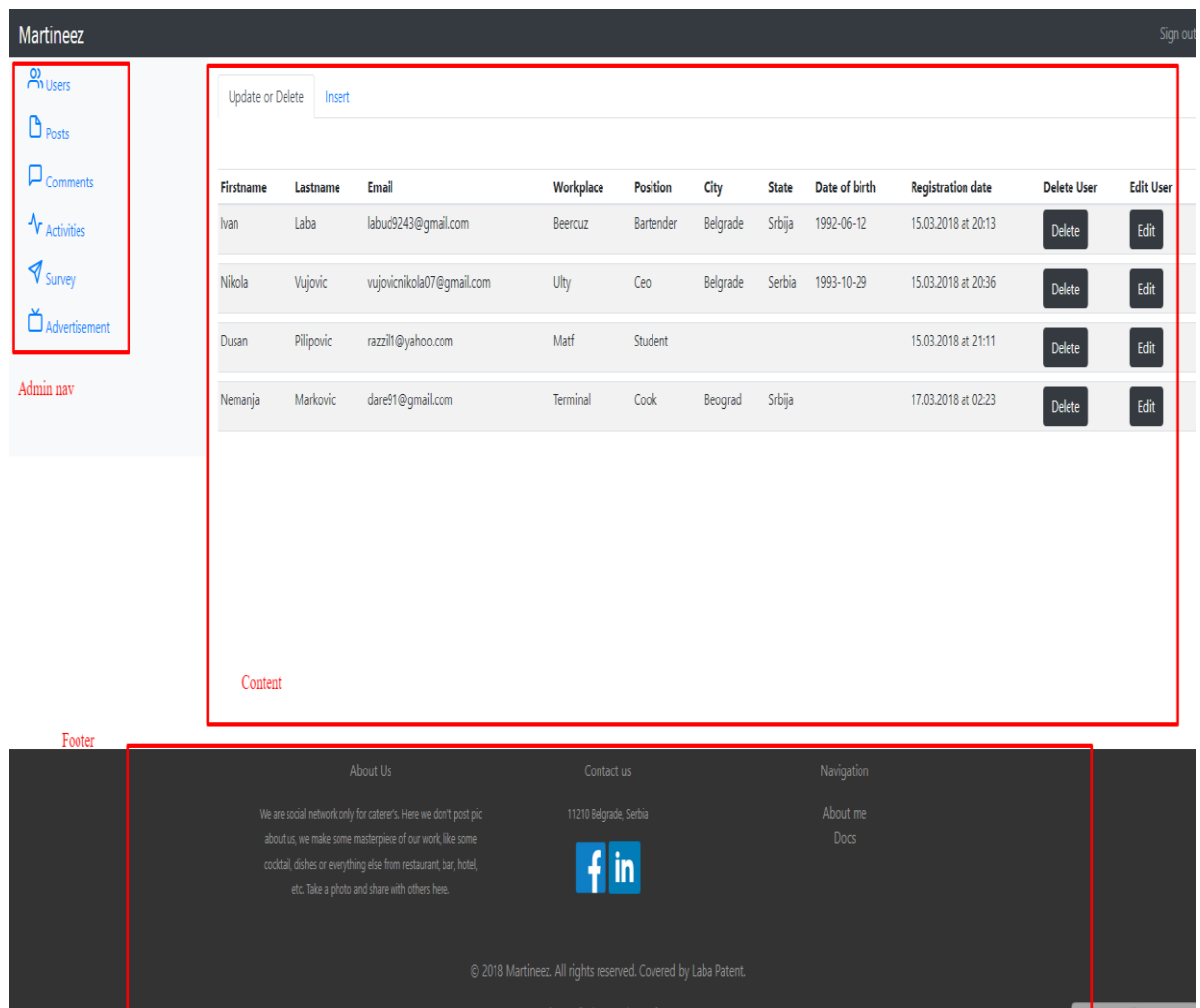
Sajt ima sledeće funkcionalnosti. Korisnik mora da se registruje da bi pristupio glavnom sadržaju sajta. Prilikom registracije kreira mu se profil koji može da uređuje. Klikom na profilnu sliku otvara se modal, koji mu dozvoljava da promeni profilnu sliku. U odeljku settings može uređivati podatke o sebi, popunjavanjem forme i klikom na edit dugme.

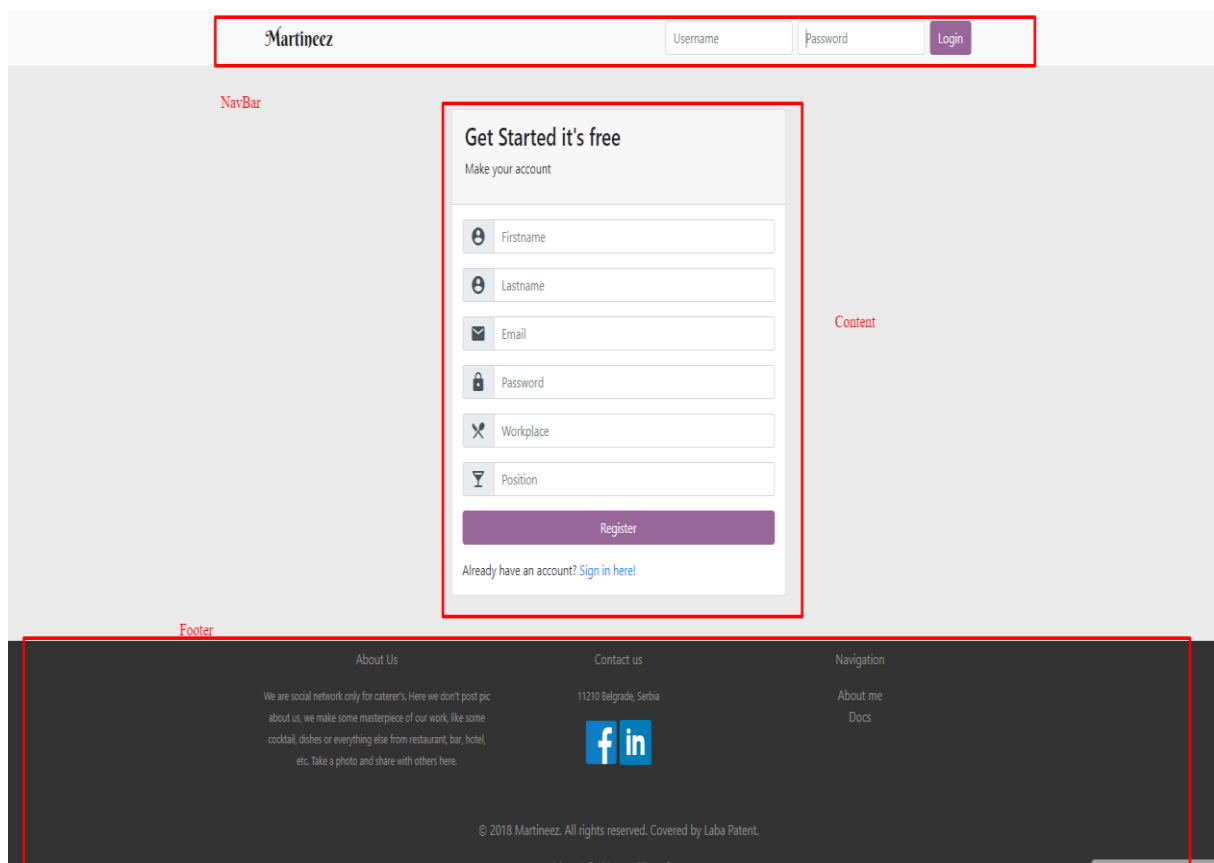
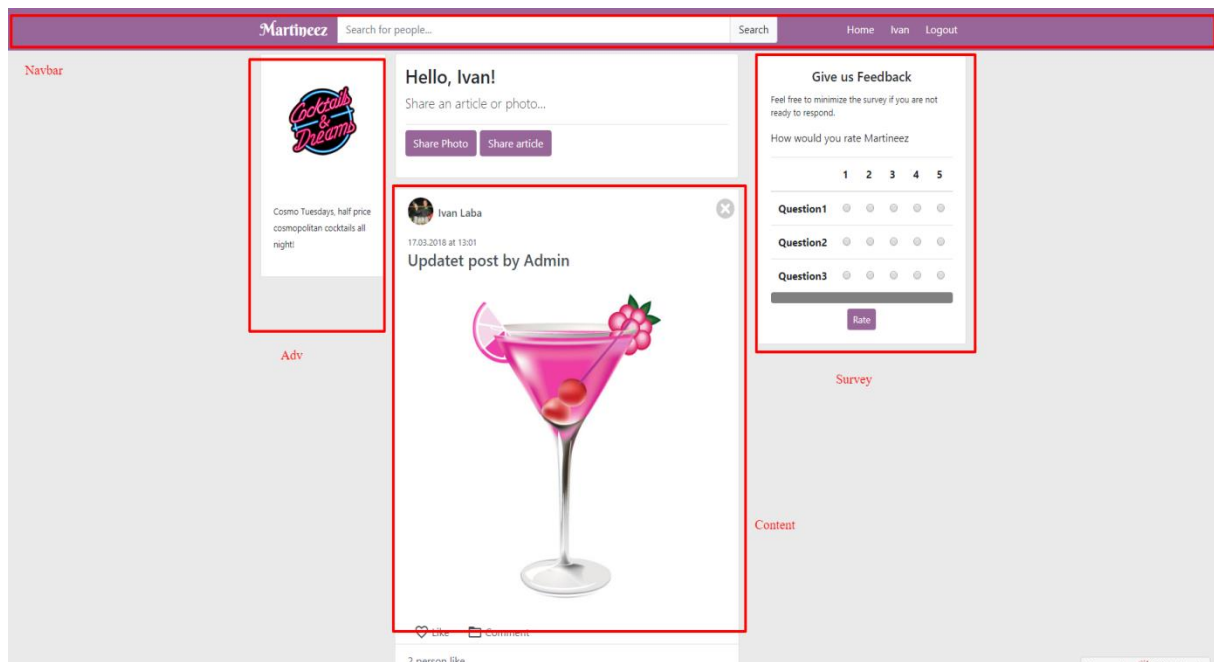
Sledeća funkcionalnost autorizovanog korisnika je deljenje slike sa naslovom ili bez, i izborom da li će podeliti sa svima ili samo u svoju galeriju. Korisnik može podeliti tekstualni post koji ostali korisnici mogu videti na home strani. Korisnik može komentarisati i lajkovati postove drugih korisnika. Unošenjem imena u search polje i klikom na dugme search, otvara se profil traženog korisnika. Sav sadržaj koji je uneo(postove, artikle) korisnik može obrisati. Postoji anketa koju korisnik može samo jednom popuniti i dati svoju ocenu o sajtu.

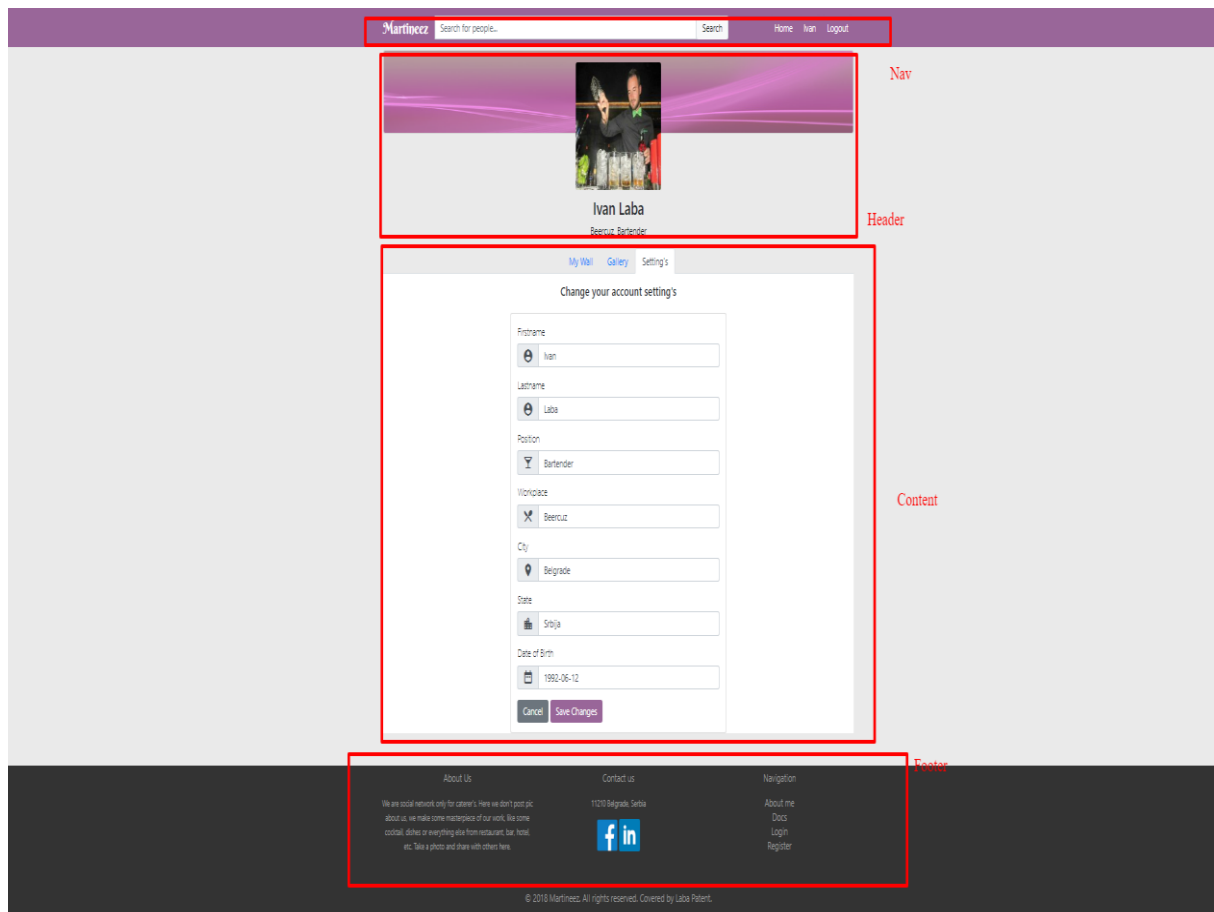
<b>Korišćeni jezici</b>	<ul style="list-style-type: none"> <li>• HTML 4 i HTML 5</li> <li>• CSS</li> <li>• JavaScript, jQuery</li> <li>• PHP</li> <li>• MySql</li> </ul>
<b>Korišćeni HTML šabloni (framework-ci)</b>	Bootstrap
<b>Korišćene biblioteke i njihovi url-ovi odakle ste ih preuzeli</b>	<ul style="list-style-type: none"> <li>• JQuery(<a href="https://code.jquery.com/jquery-3.3.1.min.js">https://code.jquery.com/jquery-3.3.1.min.js</a> )</li> <li>• Bootstrap(<a href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css</a> )</li> <li>• JQueryValidator(<a href="https://cdn.jsdelivr.net/npm/jquery-validation@1.17.0/dist/additional-methods.js">https://cdn.jsdelivr.net/npm/jquery-validation@1.17.0/dist/additional-methods.js</a>)</li> <li>• Google material icon(<a href="https://material.io/icons/">https://material.io/icons/</a>)</li> </ul>
<b>Korišćene biblioteke iz framework-a</b>	<ul style="list-style-type: none"> <li>• Laravel</li> </ul>
<b>Korišćene php biblioteke van framework-a</b>	
<b>Koje funkcionalnosti su realizovane putem AJAX-a</b>	Aneka(glasanje), Paginacija na svojoj profilnoj stranici.
<b>Navešti lokacije gde je napisan AJAX kod (fajlovi ili delovi html stranice)</b>	script.js, admin.js
<b>Navešti lokacije gde je napisan sopstveni JavaScript</b>	script.js, admin.js

<b>kod (fajlovi ili delovi html stranice)</b>	
<b>URL sajta/aplikacije</b>	<a href="http://labud9243.000webhostapp.com/">http://labud9243.000webhostapp.com/</a>
<b>Pristupni parametri</b>	Poslati na mail

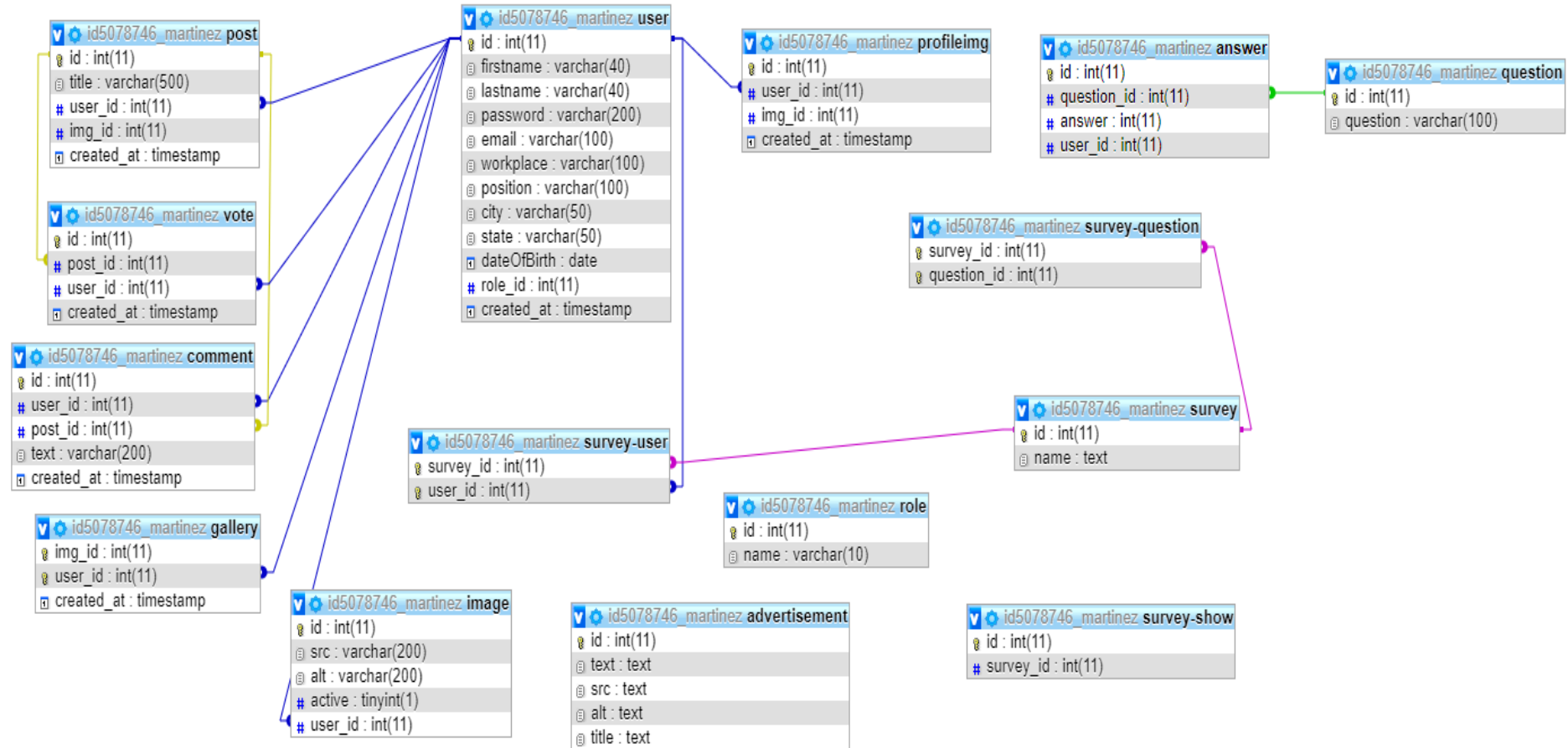
## 2. Skica struktura stranica







### 3. Dijagram baze podataka





### 3.MVC Organizacija

#### Kontroleri

AdminContoroller
__construct()
Index()
Delete()
Edit()
editUser()
registerUser()
Post()
showPost()
showUpForm()
updatePost()
ShowPostCom()
deletePost()
Comment()
showComment()
deleteComment()
updateComment()
Activities()
showActivities()
Survey()
showSurvey()
updateSurvey()
insertSurvey()
deleteSurvey()
Adv()
insertAdv()
updateAdv()
updateAdvFin()

AjaxController()
__construct()
delPost()
Like()
viewLikes()
Survey()
getSuggestion()
viewProfil()
pagination

ArticleController
__construct()
Store()

CommetnController
__construct
Store()
Show()
delComm()

HomeController
__construct()
Index()

ImgController
__construct()
Store()
editProfileImg()

IndexController
__construct
Index()
showLogin()
Login()
Logout()
Store()

ProfilController
__construct()
Index()
Edit()

## Modeli

Advertisement
getAllAdvertisement()
insertAdv()
getAdvById()
updateAdv()
Delete()

Article
Insert()

Comment()
Store()
Show()

Image
insertImg()
getProfileImg()
getGallery()

Post
insertPost()
getAllPost()
getAllPostById()
getAllPostForPagination()
getAllForUser

Survey
getSurveyById()
GetStatistic()
getNumUser()
updateQuestion()
insertSurvey()
insertQuestion()
insertQuestionSurvey()
deleteSurvey()
getSurveyShow()

User
checkMail()
getAllUsers()
newUser()
Login()
getUserById()
editUser()
regUser()
Vote
insertLike()
checkIfLiked()
Unlike()
showNumLikes()
Likers()

## View

Views/app
Admin-layout.blade.php
Index.blade.php
Layout.blade.php

Views/components
Advertisement.blade.php
Article-modal.blade.php
c-form.blade.php
First-time.blade.php
First-time.blade.php
Footer.blade.php
Gallery.blade.php
Img-post.blade.php
Modal.blade.php
profileModal.blade.php
Survey.blade.php
profileModal.blade.php
Text-post.blade.php
Update-form.blade.php

Views/pages
Author.blade.php
<a href="#">Home.blade.php</a>
Login.blade.php
Profil.blade.php
Test.blade.php

Views/admin-pages
Activities.blade.php
Advertisement.blade.php
Comment.blade.php
Post.blade.php
Survey.blade.php
User.blade.php

## 4. javaScript kod

```

$(document).ready(function () {

    $('#btn-edit').on('click', function (e) {
        e.preventDefault();
        var url = window.location.href+"/edit";
        var id = $(this).attr('rel');
        $.ajax({
            type: "GET",
            url: url,
            data: {id: id},
            success: function (data) {
                $('#'+id).html(data);
            }, error: function(error){
                console.log(error);
            }
        })
    });

    $('#ddl-cpost').change(function (e) {
        e.preventDefault();
        const id = $(this).val();
        var url = window.location.href+"/show";
        if(id!=='0'){
            $.ajax({
                type: "GET",
                url: url,
                data: {id: id},
                success: function (data) {
                    $('#prikaz').html(data);
                },error: function (error) {
                    console.log(error);
                }
            })
        }
    });

    $('#ddl-cpostCom').change(function (e) {
        e.preventDefault();
        const id = $(this).val();
        var url = window.location.href+"/show/post";
        if(id!=='0'){
            $.ajax({
                type: "GET",
                url: url,
                data: {id: id},
                success: function (data) {
                    $('#prikaz').html(data);
                },error: function (error) {
                    console.log(error);
                }
            })
        }
    });

    $(document).on('click', '.btn-delete-post', function (e) {
        e.preventDefault();
        var id = $(this).attr('rel');
        var url = window.location.href+"/delete";
        $.ajax({
            type: "GET",
            url: url,
            data: {id: id},
            success: function () {
                location.reload();
            }, error: function (error) {
                console.log(error);
            }
        })
    });
});

```

```

});

$(document).on('click', '.btn-update-post', function (e) {
    e.preventDefault();
    var id = $(this).attr('rel');
    var url = window.location.href+"/showUpdate";
    $.ajax({
        type: "GET",
        url: url,
        data: {id: id},
        success: function (data) {
            $('#prikaz').html(data);
        }, error: function (error) {
            console.log(error);
        }
    })
});

$(document).on('click', '.btn-comment-post', function (e) {
    e.preventDefault();

    var id = $(this).attr('rel');
    var url = window.location.href+"/insertComment";
    var comment = "<div id='new'+id+'\" class='list-group-item list-group-item-
action flex-column align-items-start pb-0 mt-2'>";
    comment += "<form action='"+url+"' method='get' id='commForm'>";
    comment += "<input type='text' name='tbComment' class='form-control mb-2'
placeholder='Enter comment' required autofocus>";
    comment += "<input type='hidden' value='"+id+"' name='id'/>";
    comment += "</br><button type='submit' class='btn btn-primary btn-sm mb-
2'>Comment</button>";
    comment += "</form>";
    comment += "</div>";
    if(document.getElementById("new"+id) === null)
    {
        $( comment ).insertAfter( this );
    }
});

$('.ddl-comment').on('change', function (e) {
    e.preventDefault();
    var id = $(this).val();
    var url = window.location.href+"/show";
    if(id!=='0'){
        $.ajax({
            type: "GET",
            url: url,
            data: {id: id},
            success: function (data) {
                $('#shComm').html(data);
            }, error: function (error) {
                console.log(error);
            }
        })
    }
});

$(document).on('click', '.btn-del-comment', function (e) {
    e.preventDefault();
    var id = $(this).attr('rel');
    var url = window.location.href+"/delete";
    $.ajax({
        type: "GET",
        url: url,
        data: {id: id},
        success: function () {
            location.reload();
        }
    })
});

```

```

        }, error: function (error) {
            console.log(error);
        }
    })
});

$(document).on('click', '.btn-upd-comment', function (e) {
    e.preventDefault();
    var id = $(this).attr('rel');
    var value = $('#com'+id).text();
    if ($('#elementID'+id).length) {
        $('#<div class="elementID'+id+'"><div class="form-group"><label
for="exampleInputPassword1">Update Comment</label><input type="text" class="form-
control" id="text'+id+'" value="'+value+'"><button type="button" class="btn btn-
primary btn-sm mt-2 btn-upd"
rel="'+id+'">Update</button></div></div>').insertAfter('#'+id);
        $('#.btn-upd').on('click', function (e) {
            e.preventDefault();
            var id = $(this).attr('rel');
            var newText = $('#text'+id).val();
            var url = window.location.href+"/updateComment";
            $.ajax({
                type: "GET",
                url: url,
                data: {id: id, txt: newText},
                success: function () {
                    location.reload();
                }, error: function (error) {
                    console.log(error);
                }
            })
        })
    }
});

$('#ddl-act').on('change', function (e) {
    e.preventDefault();
    var id = $(this).val();
    var url = window.location.href+"/show";
    if(id!=='0'){
        $.ajax({
            type: "GET",
            url: url,
            data: {id: id},
            success: function (data) {
                $('#showAct').html(data);
            }, error: function (error) {
                console.log(error);
            }
        })
    }
});

$('#ddl-survey').on('change', function (e) {
    var id = $(this).val();
    var url = window.location.href+"/show";
    if(id!=='0')
    {
        $.ajax({
            type: "GET",
            url: url,
            data: {id: id},
            success: function (data) {
                $('#shQuest').html(data);
            }, error: function (error) {
                console.log(error);
            }
        })
    }
});

```

```

    });

$(document).on('click', '.btn-up-ques', function (e) {
    e.preventDefault();
    var url = window.location.href + "/update";
    var id = $(this).attr('rel');
    if (!$('#in'+id).length){
        $( "<form action='"+url+"' method='get'><input id='in'+id+' "
name='tbQuestion' type='text' class='form-control my-2'><input type='hidden'
name='idQuestion' value='"+id+"'><button type='submit' class='btn btn-dark btn-sm
btn-up-q'>Change</button></form>" ).insertAfter(this);
    }
});

$(document).on('submit', '#insertSurvey', function (e) {
    var name = $("input[name='name']").val();
    var q1 = $("input[name='question1']").val();
    var q2 = $("input[name='question2']").val();
    var q3 = $("input[name='question3']").val();
    if(q1 === "" || q2 === "" || q3 === "" || name === ""){
        alert("Morate popuniti sva polja");
        e.preventDefault();
    }
});

$('#insertNewSurvey').on('submit', function (e) {
    var id = $('#ddlDel :selected').val();
    if(id === '0'){
        e.preventDefault();
        alert("Morate odabrati anketu!")
    }
});

$('#insertAdv').on('submit', function (e) {
    var text = $("input[name='text']").val();
    var title = $("input[name='title']").val();
    var image = $("input[name='image']").val();
    if(text === "" || title === "" || image === ""){
        e.preventDefault();
        alert("Morate popuniti sva polja!")
    }
});

$('#ddl-upd-adv').on('change', function (e) {
    var id = $(this).val();
    if(id !== '0'){
        var url = window.location.href + "/update";
        var _token = $("input[name='_token']").val();
        $.ajax({
            url: url,
            type: "POST",
            data: {id: id, _token: _token},
            success: function (data) {
                $('#showUpdAdv').html(data);
            }, error: function(error){
                console.log(error);
            }
        })
    }
    }else{
        e.preventDefault();
    }
});

$.validator.addMethod(
    "regex",
    function(value, element, regexp) {
        var check = false;

```



```

        return this.optional(element) || regexp.test(value);
    },
    "Please check your input."
);

$('#adminInsert').validate({
    rules: {
        tbFirstname: {
            required: true,
            minlength: 3
        },
        tbLastname: {
            required: true,
            minlength: 3
        },
        tbPosition: {
            required: true,
            minlength: 3
        },
        tbWorkplace: {
            required: true,
            minlength: 3
        },
        tbCity: {
            required: true,
            minlength: 3
        },
        tbState: {
            required: true,
            minlength: 3
        },
        tbMail: {
            email: true,
            required: true,
            minlength: 3
        },
        tbPassword: {
            required: true,
            minlength: 3,
            regex: /^(?=.*\d)(?=.*[a-zA-Z])[a-zA-Z0-9]{7,}$/
        }
    },
    messages: {
        tbFirstname: {
            required: 'Enter firstname',
            minlength: jQuery.validator.format("At least {0} characters
required!")
        },
        tbLastname: {
            required: 'Enter lastname',
            minlength: jQuery.validator.format("At least {0} characters
required!")
        },
        tbWorkplace: {
            required: 'Enter workplace',
            minlength: jQuery.validator.format("At least {0} characters
required!")
        },
        tbPosition: {
            required: 'Enter position',
            minlength: jQuery.validator.format("At least {0} characters
required!")
        },
        tbCity: {
            required: 'Enter city',
            minlength: jQuery.validator.format("At least {0} characters
required!")
        },
    },
});

```

```

        tbState: {
            required: 'Enter state',
            minlength: jQuery.validator.format("At least {0} characters
required!")
        },
        tbPassword: {
            required: 'Enter password',
            minlength: jQuery.validator.format("At least {0} characters
required!"),
            regex: 'At least 4 letters and 3 digit!'
        },
        tbMail: {
            required: 'Enter mail',
            minlength: jQuery.validator.format("At least {0} characters
required!"),
        }
    }

});

});
$.ready( function() {

    //ADD NEW METHOD FOR REGEX
    $.validator.addMethod(
        "regex",
        function(value, element, regexp) {
            var check = false;
            return this.optional(element) || regexp.test(value);
        },
        "Please check your input."
    );

    //ADD NEW METHOD FOR IMAGE SIZE
    $.validator.addMethod("filesize_max", function(value, element, param) {
        var isOptional = this.optional(element),
            file;

        if(isOptional) {
            return isOptional;
        }

        if ($(element).attr("type") === "file") {

            if (element.files && element.files.length) {

                file = element.files[0];
                return ( file.size && file.size <= param );
            }
        }
        return false;
    }, "File size is too large. Max 2MB!");

    //regex for fname, lname and pos!
    var reg = /^[a-zA-Z]+(([',. -][a-zA-Z ])?[a-zA-Z]*)*$/;

    //CHECK LOGIN FORM
    $("#log-form").validate({
        rules: {
            tbMail: {
                email: true,
                required : true
            },
            tbPassword: {
                required :true,
                regex : /^(?=.*\d) (?=.*[a-zA-Z]) [a-zA-Z0-9]{7,}$/
            }
        },
        messages: {

```

```

        tbMail: {
            required: 'required',
            mail: 'valid mail'
        },
        tbPassword: {
            required: 'required',
            regex: 'valid password'
        }
    },
    errorPlacement: function(error, element) {
        var el = element.attr('name').toLowerCase();
        var res = el.substring(2);
        var name = '#' + res;
        error.addClass('alert alert-danger');
        error.insertAfter(name);
    }
});

//CHECK ARTICLE FORM
$('#article-form').validate({
    rules: {
        tbArticle: {
            required: true,
            minlength: 3,
        }
    },
    messages: {
        tbArticle: {
            required: 'Textbox cannot be empty',
            minlength: jQuery.validator.format("At least {0} characters required!")
        }
    }
});

//CHECK POST IMG
$('.img-form').validate({
    rules: {
        tbTitle: {
            required: false,
            regex: /^[_A-z0-9]*((-|\s)*[_A-z0-9])*$$/
        },
        image: {
            required: true,
            extension: "jpg,jpeg,png,gif",
            filesize_max : 2400000
        }
    },
    messages: {
        tbTitle: 'Only letters and numbers!',
        image: {
            extension: 'only jpg, jpeg, gif, png',
            required: 'required image'
        }
    }
});

//CHECK PROFILE IMAGE
$('#img-prf').validate({
    rules: {
        image: {

```

```

        required: true,
        extension: "jpg,jpeg,png,gif",
        filesize_max : 2400000
    },
    },
    messages: {

        image: {
            extension: 'only jpg, jpeg, gif, png',
            required: 'required image'
        }
    }
}

});

//CHECK FILE SIZE
// $('#img-form').submit(function () {
//
//     var s = $("#fileimg")[0].files[0].size;
//     alert(s);
//
//
//
// });

//CHECK REGISTER FORM
$("#register-form").validate({
    rules: {

        tbFirstname: {
            minlength: 2,
            required: true,
            regex: reg
        },
        tbLastname: {
            minlength: 2,
            required: true,
            regex: reg
        },
        tbMail: {
            email: true,
            required: true
        },
        tbPassword:{
            required: true,
            regex : /^(?=.*\d) (?=.*[a-zA-Z]) [a-zA-Z0-9]{7,}$/
        },
        tbPosition: {
            required: true,
            regex: reg,
            minlength: 3
        },
        tbWorkplace: {
            required: true,
            regex: /^[_A-z0-9]*((-|s)*[_A-z0-9])*$$/,
            minlength: 3
        },
        tbCity: {
            minlength: 3,
            regex: reg
        },
        tbState: {
            minlength: 3,
            regex: reg
        }
    },
    messages: {

```

```

        tbFirstname: {
            required: "We need your firstname!",
            minlength: jQuery.validator.format("At least {0} characters
required!"),
            regex: "Only letters!"
        },
        tbLastname: {
            required: "We need your lastname!",
            minlength: jQuery.validator.format("At least {0} characters
required!"),
            regex: "Only letters!"
        },
        tbMail: {
            required: "We need your email address to contact you!",
            email: "Enter valid address!"
        },
        tbPassword: {
            required: "Password is required!",
            regex: "Minimum 7 character length, only digits and letters!"
        },
        tbPosition: {
            required: "Position is required!",
            regex: "Only letters",
            minlength: jQuery.validator.format("At least {0} characters
required!"),
        },
        tbWorkplace: {
            required: "If you are unemployed, set unemployed!",
            regex: "Only letters and number!",
            minlength: jQuery.validator.format("At least {0} characters
required!"),
        },
        tbCity: {
            minlength: jQuery.validator.format("At least {0} characters
required!"),
            regex: 'Only letters'
        },
        tbState: {
            minlength: jQuery.validator.format("At least {0} characters
required!"),
            regex: 'Only letters'
        }
    },
    errorPlacement: function(error, element) {
        var el = element.attr('name').toLowerCase();
        var res = el.substring(2);
        var name = '#' + res;
        error.addClass('alert alert-danger');
        error.insertAfter(name);
    }
});

});
if ($('#back-to-top').length) {
    var scrollTrigger = 100, // px
        backToTop = function () {
            var scrollTop = $(window).scrollTop();
            if (scrollTop > scrollTrigger) {
                $('#back-to-top').addClass('show');
            } else {
                $('#back-to-top').removeClass('show');
            }
        };
    backToTop();
    $(window).on('scroll', function () {
        backToTop();
    });
}

```

```

    $('#back-to-top').on('click', function (e) {
        e.preventDefault();
        $('html,body').animate({
            scrollTop: 0
        }, 700);
    });
}
$(document).ready( function() {

    $("#search-form").on('submit', function (e) {
        e.preventDefault();

        var id = $("input[name='tbSearch']").attr('rel');
        if(id!=null){
            var url = window.location.href+"/viewProfil";
            $.ajax({
                type: "GET",
                url: url,
                data: {id: id},
                success: function (data) {
                    $('#profilPrikaz').html(data);
                    console.log(data);
                }, error: function(error){
                    console.log(error);
                }
            })
        }
    });

    $(document).on('click', '.btnDel', function (e) {
        e.preventDefault();
        if(confirm('Delete post?')){
            var url = window.location.href+"/deletePost";
            var id = $(this).attr('rel');
            $.ajax({
                url: url,
                type: "GET",
                data: {id: id},
                success: function () {
                    location.reload();
                },error: function (error) {
                    console.log(error);
                }
            })
        }
    });

    load_data(1);
    function load_data(page){
        var url = window.location.href+"/pag";
        $.ajax({
            url: url,
            method: "GET",
            data: {page: page},
            success: function (data) {
                $('#pagination_data').html(data);
            }, error: function (error) {
                console.log(error);
            }
        })
    }
    $(document).on('click', '.pagination_link', function () {
        var page = $(this).attr('id');
        load_data(page);
    });
});

```

```

$(document).on('click', '.btnDelComm', function (e) {
    e.preventDefault();
    if(confirm('Delete comment?')){
        var url = window.location.href+"/deleteComment";
        var id = $(this).attr('rel');
        $.ajax({
            url: url,
            type: "GET",
            data: {id: id},
            success: function () {
                location.reload();
            },error: function (error) {
                console.log(error);
            }
        });
    }
});

var progres = 0;

$("input[name=rbQuestion1]").click(function () {
    progres +=33;
    $('.progress').val(progres);
});
$("input[name=rbQuestion2]").click(function () {
    progres +=33;
    $('.progress').val(progres);
});
$("input[name=rbQuestion3]").click(function () {
    progres +=34;
    $('.progress').val(progres);
});

$('#form-survey').on('submit', function (e) {
    e.preventDefault();
    var q1 = $('input[name=rbQuestion1]:checked').val();
    var q2 = $('input[name=rbQuestion2]:checked').val();
    var q3 = $('input[name=rbQuestion3]:checked').val();
    if(q1 == null || q2 == null || q3 == null)
    {
        var div = document.createElement("DIV");
        div.innerHTML = "<div id='errorMsg' class='alert alert-danger'>Check all the answers!</div>";
        $('#error-survey').html(div);
        $('#errorMsg').delay(2000).fadeOut(400);
    }else{
        var url = window.location.href+"/subSurvey";
        $.ajax({
            type: 'post',
            url: url,
            data: $('#form-survey').serialize(),
            success: function (data) {
                if(data.trim()==''){
                    $('#survey-div').fadeOut("normal", function () {
                        $(this).remove();
                    });
                }else{
                    var div = document.createElement("DIV");
                    div.innerHTML = "<div id='errorMsg' class='alert alert-danger'>You already vote, Thank's!</div>";
                    $('#error-survey').html(div);
                    $('#errorMsg').delay(2000).fadeOut(400);
                }
            }
        });
    }
});

```

```

    });
}
});

$(document).on('click', '.commPost', function () {
    var id = $(this).attr('rel');
    var url = window.location.href+"/insertComment";
    var comment = "<div class='list-group-item list-group-item-action flex-
column align-items-start pb-0'>";
    comment += "<form action="+url+" method='get' id='commForm'>";
    comment += "<input type='text' name='tbComment' class='form-control mb-2'
placeholder='Enter comment' required autofocus>";
    comment += "<input type='hidden' value="+id+" name='id'/>";
    comment += "</br><button type='submit' class='btn btn-primary btn-sm mb-
2'>Comment</button>";
    comment += "</form>";
    comment += "</div>";
    $('#'+id+"comm").html(comment);
    //CHECK COMMENT FORM
    $('#commForm').validate({
        rules: {
            tbComment: {
                required: true,
                minlength: 3,
            }
        },
        messages: {
            tbComment: {
                required: 'Textbox cannot be empty',
                minlength: jQuery.validator.format("At least {0} characters
required!")
            }
        }
    });
});

$(document).on('click', '.viewComment', function (e) {
    e.preventDefault();
    var id = $(this).attr('rel');
    var url = window.location.href+"/showLikers";
    $.ajax({
        type: "GET",
        url: url,
        data: {id: id},
        dataType: 'json',
        success: function (data) {
            $('#'+id+'showComm').html(data);
        }, error: function (data) {
            console.log(data);
        }
    });
});

$(document).on('click', '.ajax-link', function(e) {
    e.preventDefault();
    var id = $(this).attr('rel');
    var url = window.location.href+"/like";
    $.ajax({
        type: "GET",
        url: url,
        data : { id: id },
        dataType: "json",
        success: function (podaci) {
            var tekst = "";
            $.each(podaci, function (index, podatak) {
                tekst += podatak.likes+" person like";
            });
        }
    });
});

```



```

        $('#'+id).html(tekst);
    }
    ,
    error: function (data) {

        console.log(data);
    }
    });
});

$(document).on('click', '.viewLikes', function (e) {
    e.preventDefault();
    var id = $(this).attr('rel');
    var url = window.location.href+"/viewLikes";
    $.ajax({
        type: "GET",
        url: url,
        data : { id: id },
        dataType : "json",
        success: function(podaci){
            var tekst = "<div id='kill' class='card text-center'>";
            tekst += "<div class='card-header'>";
            tekst += "Likes";
            tekst += "</div>";
            tekst += "<div class='card-body'><p class='card-text'>";
            $.each(podaci, function(index, podatak){
                tekst += podatak.firstname + " " + podatak.lastname + "</br>";
            });
            tekst += "</p>";
            tekst += "<button class='btn btn-secondary seeLikes'>Close</button>";
            tekst += "</div></div>";
            $('body').append(tekst);
            $('.seeLikes').click(function () {
                $('#kill').remove();
            });
        },error: function (data) {
            console.log(data);
        }
    });
});

$(".fade-out").fadeIn(3000, function() {
    setTimeout(function() {
        $(".fade-out").fadeOut("slow");
    },4000);
});
});

```

## 5. PHP kod

### AdminController

```

<?php

namespace App\Http\Controllers;

use App\Models\Advertisement;

```

```

use App\Models\Survey;
use App\Models\User;
use App\Rules\ddlRule;
use Carbon\Carbon;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Log;
use Intervention\Image\ImageManagerStatic as img;

class AdminController extends Controller
{
    private $data;
    private $model;

    //region user methods
    public function index(){
        $this->model = new User();
        try{
            $this->data['users'] = $this->model->getAllUsers();
            return view('pages.admin-pages.user',$this->data);
        }catch (\Exception $e){
            Log::error($e);
            return 'Sorry! Something is wrong with database!';
        }
    }

    public function delete($id){
        try{
            DB::table('user')
                ->where('id','=',$id)
                ->delete();
        }catch (\Exception $e){
            Log::error($e);
            return 'Sorry! Something is wrong with database!';
        }
        return redirect()->back();
    }

    public function edit(Request $request){
        $id = $request->get('id');
        $url = url()->current()."/update/".$id;
        $this->model = new User();
        try{
            $this->data = $this->model->getUserById($id);
            $output = "<form action='".$url.'" method='post'>".csrf_field()."<div
class='form-group row'><label for='tbFirstname' class='col-sm-2 col-form-
label'>Firstname:</label><div class='col-sm-10'><input type='text' class='form-
control' name='tbFirstname' value='".$this->data->firstname.'"></div></div><div
class='form-group row'><label for='tbLastname' class='col-sm-2 col-form-
label'>Lastname:</label><div class='col-sm-10'><input type='text' class='form-
control' name='tbLastname' value='".$this->data->lastname.'"></div></div><div
class='form-group row'><label for='tbPosition' class='col-sm-2 col-form-
label'>Position:</label><div class='col-sm-10'><input type='text' class='form-
control' name='tbPosition' value='".$this->data->position.'"></div></div><div
class='form-group row'><label for='tbWorkplace' class='col-sm-2 col-form-
label'>Workplace:</label><div class='col-sm-10'><input type='text' class='form-
control' name='tbWorkplace' value='".$this->data->workplace.'"></div></div><div
class='form-group row'><label for='tbCity' class='col-sm-2 col-form-
label'>City:</label><div class='col-sm-10'><input type='text' class='form-control'
name='tbCity' value='".$this->data->city.'"></div></div><div class='form-group
row'><label for='tbState' class='col-sm-2 col-form-label'>State:</label><div
class='col-sm-10'><input type='text' class='form-control' name='tbState'
value='".$this->data->state.'"></div></div><div class='form-group row'><label
for='date' class='col-sm-2 col-form-label'>Date of Birth:</label><div class='col-
sm-10'><input type='date' class='form-control' name='date'></div></div><div
class='form-group row'><div class='col-sm-10'><button type='submit' class='btn btn-
dark'>Confirm</button></div></div></form>";

```

```

        return $output;
    } catch (\Exception $e) {
        Log::error($e);
        return 'Sorry! Something is wrong with database!';
    }
}

}

public function editUser(Request $request, $id) {
    $fname = $request->get('tbFirstname');
    $lname = $request->get('tbLastname');
    $pos = $request->get('tbPosition');
    $work = $request->get('tbWorkplace');
    $city = $request->get('tbCity');
    $state = $request->get('tbState');
    $date = $request->get('date');
    try {
        $this->model = new User();
        $this->model->setFirstname($fname);
        $this->model->setLastname($lname);
        $this->model->setPosition($pos);
        $this->model->setWorkplace($work);
        $this->model->setCity($city);
        $this->model->setState($state);
        $this->model->setDateOfBirth($date);
        $this->model->editUser($id);
        return redirect()->back();
    } catch (\Exception $e) {
        Log::error($e);
        return 'Sorry! Something is wrong with database!';
    }
}

public function registerUser(Request $request) {
    //CHECKING EMAIL
    $this->model = new User();
    $mail = $request->input('tbMail');
    $check = $this->model->checkMail($mail);
    if ($check) {
        return redirect()->back()->with('msg', 'There is already user with this email address!');
    }
    else {
        try {
            $this->model->setFirstname($request->input('tbFirstname'));
            $this->model->setLastname($request->input('tbLastname'));
            $pass = Hash::make($request->input('tbPassword'));
            $this->model->setPassword($pass);
            $this->model->setEmail($request->input('tbMail'));
            $this->model->setWorkplace($request->input('tbWorkplace'));
            $this->model->setPosition($request->input('tbPosition'));
            $this->model->setCity($request->input('tbCity'));
            $this->model->setState($request->input('tbState'));
            $this->model->setDateOfBirth($request->input('date'));
            $this->model->regUser();
            return redirect()->back();
        } catch (\Exception $e) {
            Log::error($e);
            return 'Sorry! Something is wrong with database!';
        }
    }
}

}

//endregion

//region post methods

```

```

public function post(){
    try{
        $this->data['users'] = DB::table('user')->get();
        return view('pages.admin-pages.post',$this->data);
    }catch (\Exception $e){
        Log::error($e->getMessage());
        return "Sorry! Something went wrong with database!";
    }
}

public function showPost(Request $request){
    $id = $request->get('id');
    try{
        $res = DB::table('post')
            ->join('user','user.id','=','post.user_id')
            ->leftJoin('image','image.id','=','post.img_id')
            ->select('post.created_at','post.title','image.src','post.id')
            ->where('post.user_id','=',$id)
            ->orderBy('created_at','desc')
            ->get();
        if(count($res)>=1){
            $out="<div class='row'>";
            foreach ($res as $item){
                if($item->src==""){
                    $out .= "<div class='card w-75'><div class='card-body'><h5
class='card-title'>Published ". Carbon::parse($item->created_at)->format('d.m.Y
\a\t H:i')."</h5><p class='card-text'>$item->title</p><button class='btn btn-
primary btn-delete-post' rel='$item->id'>Delete</button><button class='btn btn-
primary btn-update-post ml-2' rel='$item->id'>Update</button></div></div>";
                }else {
                    $out .= "<div class='card w-75'><div class='card-body'><h5
class='card-title'>Published ". Carbon::parse($item->created_at)->format('d.m.Y
\a\t H:i')."</h5><p class='card-text'>$item->title</p><img class='card-img'
style='max-height: 600px' src='".asset('').$item->src."><button class='btn btn-
primary mt-3 btn-delete-post' rel='$item->id'>Delete</button><button class='btn
btn-primary mt-3 btn-update-post ml-2' rel='$item-
>id'>Update</button></div></div>";
                }
            }
            $out.="</div>";
            return $out;
        }else{
            $out="<div class='alert alert-danger'>This user has no post</div>";
            return $out;
        }
    }catch (\Exception $e){
        Log::error($e->getMessage());
        return "Sorry! Something went wrong with database!";
    }
}

public function showUpForm(Request $request){
    $id = $request->get('id');
    try{
        $res = DB::table('post')
            ->join('user','user.id','=','post.user_id')
            ->leftJoin('image','image.id','=','post.img_id')
            ->select('post.created_at','post.title','image.src','post.id')
            ->where('post.id','=',$id)
            ->orderBy('created_at','desc')
            ->get();
        $out="<div class='row'>";
        foreach ($res as $item){
            if($item->src==""){
                $out .= "<div class='card w-75'>
<div class='card-body'>

```



```

primary mt-3 btn-comment-post' rel='$item->id'>Comment</button></div></div><div
id='". $item->id. "com"></div>";
    }
    }
    $out.="</div>";
    return $out;
} else {
    $out="<div class='alert alert-danger'>This user has no post</div>";
    return $out;
}

} catch (\Exception $e) {
    Log::error($e->getMessage());
    return "Sorry! Something went wrong with database!";
}
}

public function deletePost(Request $request){
    try{
        $id = $request->get('id');
        DB::table('post')
            ->where('id', '=', $id)
            ->delete();
    } catch (\Exception $e) {
        Log::error($e->getMessage());
        return "Sorry! Something went wrong with database!";
    }
}

//endregion

//region comment methods
public function comment() {
    $this->data['users'] = DB::table('user')->get();
    return view('pages.admin-pages.comment', $this->data);
}

public function showComment(Request $request){
    $id = $request->get('id');
    try{
        $res = DB::select('SELECT c.id, c.text, c.created_at, u.firstname,
u.lastname FROM (comment c JOIN post p ON c.post_id = p.id) JOIN user u ON u.id =
p.user_id WHERE c.user_id = ?', [$id]);
        if(count($res)>=1){
            $output = "<div class='row'>";
            foreach ($res as $item){
                $output .="<div class='card w-75'><div class='card-body'
id='$item->id'><h5 class='card-title'>Commented ". Carbon::parse($item-
>created_at)->format('d.m.Y \a\t H:i')."<small> at $item->firstname $item-
>lastname post</small></h5><p class='card-text' id='com$item->id'>$item-
>text</p><button class='btn btn-primary btn-del-comment' rel='$item-
>id'>Delete</button><button class='btn btn-primary btn-upd-comment ml-3'
rel='$item->id'>Update</button></div></div>";
            }
            $output.="</div>";
            return $output;
        } else {
            $output = "<div class='alert alert-danger'>This user has no
comments!</div>";
            return $output;
        }
    } catch (\Exception $e) {
        Log::error($e->getMessage());
        return "Sorry! Something went wrong with database!";
    }
}

public function deleteComment(Request $request){
    try{

```

```

        $id = $request->get('id');
        DB::table('comment')
            ->where('id', '=', $id)
            ->delete();
    } catch (\Exception $e) {
        Log::error($e->getMessage());
        return "Sorry! Something went wrong with database!";
    }
}

public function updateComment(Request $request) {
    $id = $request->get('id');
    $text = $request->get('txt');
    try {
        DB::table('comment')
            ->where('id', '=', $id)
            ->update(['text' => $text]);
    } catch (\Exception $e) {
        Log::error($e->getMessage());
        return "Sorry! Something went wrong with database!";
    }
}

//endregion

//region user activities
public function activities() {
    $this->data['users'] = DB::table('user')->get();
    return view('pages.admin-pages.activities', $this->data);
}

public function showActivities(Request $request) {
    $id = $request->get('id');
    try {
        $regDate = DB::table('user')
            ->select('created_at', 'firstname')
            ->where('id', '=', $id)
            ->first();
        $output = "<div class='alert alert-success'>$regDate->firstname
registered on Martinez ".Carbon::parse($regDate->created_at)->format('d.m.Y \a\t
H:i')."</div>";

        $posts = DB::table('post')
            ->join('user', 'user.id', '=', 'post.user_id')
            ->where('post.user_id', '=', $id)
            ->select('post.created_at', 'post.img_id')
            ->get();
        if(count($posts)>=1) {
            $output.="<div class='d-flex justify-content-between flex-wrap
flex-md-nowrap align-items-center pb-2 mb-3 border-bottom'><h1
class='h2'>Posts</h1></div>";
            $output.="<ul class='list-group'>";
            foreach ($posts as $item) {
                if($item->img_id == "") {
                    $output.="<li class='list-group-
item'>".Carbon::parse($item->created_at)->format('d.m.Y \a\t
H:i')." $regDate->firstname posted article</li>";
                } else {
                    $output.="<li class='list-group-
item'>".Carbon::parse($item->created_at)->format('d.m.Y \a\t
H:i')." $regDate->firstname posted photo</li>";
                }
            }
            $output.="</ul>";
        }

        $res = DB::select('SELECT c.created_at, u.firstname, u.lastname FROM
(comment c JOIN post p ON c.post_id = p.id) JOIN user u ON u.id = p.user_id WHERE
c.user_id = ?', [$id]);

```

```

        if(count($res)>=1){
            $output.="<div class=\"d-flex justify-content-between flex-wrap
flex-md-nowrap align-items-center mt-3 pb-2 mb-3 border-bottom\"><h1
class=\"h2\">Comments</h1></div>";
            $output.="<ul class=\"list-group\">";
            foreach ($res as $item){
                $output.="<li class='list-group-item'>".Carbon::parse($item-
>created_at)->format('d.m.Y \a\t H:i')." $regDate->firstname commented on $item-
>firstname posts!</li>";
            }
            $output.="</ul>";
        }

        $likes = DB::select('SELECT v.created_at, u.firstname, u.lastname FROM
(vote v JOIN post p ON v.post_id = p.id) JOIN user u ON u.id = p.user_id WHERE
v.user_id = ?', [$id]);
        if(count($likes)>=1){
            $output.="<div class=\"d-flex justify-content-between flex-wrap
flex-md-nowrap align-items-center mt-3 pb-2 mb-3 border-bottom\"><h1
class=\"h2\">Likes</h1></div>";
            $output.="<ul class=\"list-group\">";
            foreach ($likes as $item){
                $output.="<li class='list-group-item'>".Carbon::parse($item-
>created_at)->format('d.m.Y \a\t H:i')." $regDate->firstname likes $item-
>firstname posts!</li>";
            }
            $output.="</ul>";
        }
        return $output;
    } catch(\Exception $e){
        Log::error($e->getMessage());
        return "Sorry! Something went wrong with database!";
    }
}
//endregion

//region survey
public function survey(){
    try{
        $this->data['surveys'] = DB::table('survey')
->get();
    } catch(\Exception $e) {
        Log::error($e->getMessage());
        return "Sorry! Something went wrong with database!";
    }
    return view('pages.admin-pages.survey', $this->data);
}

public function showSurvey(Request $request){
    $id = $request->get('id');
    $this->model = new Survey();
    $this->model->setId($id);
    try{
        $res = $this->model->getStatistic($id);
        if(count($res)>=1){
            $output = "<table class='table table-
hover'><thead><th>Question</th><th>Average rate</th><th>Statistic</th><th>Update
question</th></thead><tbody>";
            foreach ($res as $item){
                $output.="<tr><td>$item->question</td><td>$item->avg</td><td>";
                for($i =1 ; $i<=5;$i++){
                    $num = $this->model->getNumUser($item->id, $i);
                    $output.= "<ul>";
                    if($num[0]->num!=0){

```



```

        $output.= "<li>".$num[0]->num." user rate with
".$i."</li>";
    }
    $output.= "</ul>";
}
$output.="</td><td><button type=\"button\" class=\"btn btn-dark
btn-up-ques\" rel=$item->id>Update</button></td></tr>";
}
$output .="</tbody></table>";

    }else{
        $output = "<div class='alert alert-danger'>No statistic for this
survey, wait for someone to vote!</div>";
    }
    return $output;
}catch(\Exception $e){
    Log::error($e->getMessage());
    return "Sorry! Something went wrong with database!";
}
}

public function updateSurvey(Request $request){
    try{
        $id = $request->get('idQuestion');
        $text = $request->get('tbQuestion');
        $this->model = new Survey();
        $this->model->updateQuestion($text, $id);
        return redirect()->back();
    }catch (\Exception $e){
        Log::error($e->getMessage());
        return "Sorry! Something went wrong with database!";
    }
}

public function insertSurvey(Request $request){
    $q1 = $request->get('question1');
    $q2 = $request->get('question2');
    $q3 = $request->get('question3');
    $name = $request->get('name');
    try{
        $this->model=new Survey();

        $idSurvey = $this->model->insertSurvey($name);
        $idq1 = $this->model->insertQuestion($q1);
        $idq2 = $this->model->insertQuestion($q2);
        $idq3 = $this->model->insertQuestion($q3);
        $this->model->insertQuestionSurvey($idSurvey, $idq1, $idq2, $idq3);

    }catch (\Exception $e){
        Log::error($e->getMessage());
        return "Sorry! Something went wrong with database!";
    }
    return redirect()->back();
}

public function deleteSurvey(Request $request){
    $btn = $request->get('btnAction');
    $id = $request->get('ddlDelete');
    try{
        if($btn==0){
            $this->model = new Survey();
            $this->model->deleteSurvey($id);
            return redirect()->back();
        }else{
            $this->model = new Survey();
            $this->model->setSurveyActive($id);
            return redirect()->back();
        }
    }catch (\Exception $e){

```

```

        Log::error($e->getMessage());
        return "Sorry! Something went wrong with database!";
    }

}

//endregion

//region advertisement
public function adv(){
    try{
        $this->model = new Advertisement();
        $this->data['advs'] = $this->model->getAllAdvertisement();
        return view('pages.admin-pages.advertisement',$this->data);
    }catch (\Exception $e){
        Log::error($e->getMessage());
        return "Sorry! Something went wrong with database!";
    }
}

public function insertAdv(Request $request){
    $text = $request->get('text');
    $title = $request->get('title');
    $rules = [
        'image' => 'required|mimes:jpg,jpeg,png,gif|max:2400'
    ];
    $custom_messages = [
        'image.max' => 'Fajl ne sme biti veci od :max KB.',
        'image.mimes' => 'Dozvoljeni formati su: :values.'
    ];
    $request->validate($rules, $custom_messages);

    $photo = $request->file('image');
    $imagename = time().'.'.$photo->getClientOriginalExtension();
    $destinationPath = public_path('/images');
    $thumb_img = img::make($photo->getRealPath())->resize(200, 200);

    try{
        $thumb_img->save($destinationPath.'/'.$imagename,80);
        $this->model=new Advertisement();
        $this->model->setText($text);
        $this->model->setTitle($title);
        $this->model->setAlt($title);
        $this->model->setSrc('images/'.$imagename);
        $this->model->insertAdv();
        return redirect()->back();
    }
    catch (\Exception $e) {
        Log::error($e->getMessage());
        return "Sorry! Something went wrong with database!";
    }
}

public function updateAdv(Request $request){
    $id = $request->get('id');
    $this->model = new Advertisement();
    $this->model->setId($id);
    $url = url('/')."/admin/adv/update/fin";
    $output = "";
    try{
        $advs = $this->model->getAdvById();
        foreach ($advs as $item){
            $output .= "<form id='insertAdv' method='POST' action='$url'
enctype='multipart/form-data'>
                ".csrf_field()."
                <input type='hidden' name='idAdv' value='$id'/>
                <div class='form-group'>
                    <label for='Title'>Title</label>

```

```

        <input type='text' class='form-control' placeholder='Title'
name='title' value='$item->title'>
    </div>
    <div class='form-group'>
        <label for='text'>Text</label>
        <textarea class='form-control' name='text' rows='3'>$item-
>text</textarea>
    </div>
    <div class='form-group'>
        Image
        <img src='".asset('/')."$item->src' alt='$item->alt' class='img-
thumbnail'>
    </div>
    <div class='form-group'>
        <label for='image'>Change Image</label>
        <input type='file' class='form-control' name='image'>
    </div>
    <button type='submit' class='btn btn-dark'>Insert</button>
</form>";
    }
} catch (\Exception $e) {
    Log::error($e->getMessage());
    return "Sorry! Something went wrong with database!";
}
return $output;
}

public function updateAdvFin(Request $request) {
    $id = $request->get('idAdv');
    $text = $request->get('text');
    $title = $request->get('title');
    $photo = $request->file('image');
    if(isset($photo)) {
        $rules = [
            'image' => 'required|mimes:jpg,jpeg,png,gif|max:2400'
        ];
        $custom_messages = [
            'image.max' => 'Fajl ne sme biti veci od :max KB.',
            'image.mimes' => 'Dozvoljeni formati su: :values.'
        ];
        $request->validate($rules, $custom_messages);

        $imagename = time().'.'.$photo->getClientOriginalExtension();
        $destinationPath = public_path('/images');
        $thumb_img = img::make($photo->getRealPath())->resize(200, 200);

        try{
            $thumb_img->save($destinationPath.'/'.$imagename,80);
            $this->model=new Advertisement();
            $this->model->setText($text);
            $this->model->setTitle($title);
            $this->model->setAlt($title);
            $this->model->setSrc('images/'.$imagename);
            $this->model->setId($id);
            $this->model->updateAdv();
            return redirect()->back();
        }
        catch (\Exception $e) {
            Log::error($e->getMessage());
            return "Sorry! Something went wrong with database!";
        }
    } else {
        try{
            DB::table('advertisement')
                ->where('id', $id)
                ->update([
                    'text' => $text,
                    'title' => $title
                ]);
        }
    }
}

```

```

        });
        return redirect()->back();
    }catch (\Exception $e){
        Log::error($e->getMessage());
        return "Sorry! Something went wrong with database!";
    }
}

}

}

public function deleteAdv(Request $request){
    $request->validate([
        'ddlDelAdv' => new ddlRule
    ]);
    $id = $request->input('ddlDelAdv');
    $this->model = new Advertisement();
    $this->model->setId($id);
    try{
        $this->model->delete();
        return redirect()->back();
    }catch (\Exception $e){
        Log::error($e->getMessage());
        return "Sorry! Something went wrong with database!";
    }
}

}

//endregion
}

```

## AjaxController.php

```

<?php

namespace App\Http\Controllers;

use App\Models\Image;
use App\Models\Post;
use App\Models\User;
use App\Models\Vote;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use Psy\Util\Json;
use Illuminate\Support\Facades\Log;

class AjaxController extends Controller
{
    private $data;

    public function delPost(Request $request){
        try{
            $id = $request->get('id');
            DB::table('post')
                ->where('id', '=', $id)
                ->delete();
        }catch (\Exception $e){
            Log::error($e);
            return 'Sorry! Something is wrong with database!';
        }
    }

    public function like(Request $request)
    {
        $post_id = $request->get('id');
        $user_id = session()->get('user')[0]->getId();
        $vote = new Vote();
        $vote->setPost_id($post_id);
    }
}

```

```

        $vote->setUser_id($user_id);
        try{
            $lp = $vote->checkIfLiked();
            if(empty($lp)){
                $vote->insertLike();
            }else{
                $vote->unlike();
            }
            $likes = $vote->showNumLikes();
            return Json::encode($likes);
        }
        catch(\Exception $e)
        {
            Log::error($e);
            return 'Sorry! Something is wrong with database!';
        }
    }

    public function viewLikes(Request $request)
    {
        $post_id = $request->get('id');
        $vote = new Vote();
        $vote->setPost_id($post_id);
        try{
            $names = $vote->likers();
            return Json::encode($names);
        }catch(\Exception $e){
            Log::error($e);
            return 'Sorry! Something is wrong with database!';
        }
    }

    public function survey(Request $request)
    {
        $surveyId = $request->get('surId');
        $user_id = session()->get('user')[0]->getId();
        $res = DB::table('survey-user')
            ->where('user_id','=', $user_id)
            ->where('survey_id','=', $surveyId)
            ->first();
        if(isset($res))
        {
            return Json::encode('You already vote! Thanks!');
        }else{
            $q1 = $request->get('rbQuestion1');
            $q2 = $request->get('rbQuestion2');
            $q3 = $request->get('rbQuestion3');
            $q1Id = $request->get('quesId1');
            $q2Id = $request->get('quesId2');
            $q3Id = $request->get('quesId3');

            try{
                DB::table('answer')
                    ->insert([[
                        'question_id' => $q1Id,
                        'answer' => $q1,
                        'user_id' => $user_id,
                    ],[
                        'question_id' => $q2Id,
                        'answer' => $q2,
                        'user_id' => $user_id,
                    ],[
                        'question_id' => $q3Id,
                        'answer' => $q3,
                        'user_id' => $user_id,
                    ]]);
                DB::table('survey-user')

```

```

        ->insert([
            'survey_id' => $surveyId,
            'user_id' => $user_id,
        ]);
    } catch (\Exception $e) {
        Log::error($e);
        return 'Sorry! Something is wrong with database!';
    }
}

}

}

public function getSuggestion() {
    return DB::table('user')
        ->where('role_id', '=', 2)
        ->select('firstname', 'lastname', 'id')
        ->get();
}

public function viewProfil(Request $request) {
    $id = $request->get('id');
    $user = new User();
    $this->data['user'] = $user->getUserById($id);
    $img = new Image();
    $this->data['profilimg'] = $img->getProfileImg($id);
    $this->data['gallery'] = $img->getGallery($id);

    $post = new Post();
    $post->setUser_id($id);
    try {
        $this->data['post'] = $post->getAllPostById();
    } catch (\Exception $e) {
        Log::error($e);
        return 'Sorry! Something is wrong with database!';
    }
    return (string) view('pages.test', $this->data);
}

public function pagination(Request $request) {
    $page = $request->get('page');
    $record_per_page = 2;
    $start_from = ($page - 1) * $record_per_page;
    $post = new Post();
    try {
        $res = $post->getAllPostForPagination($start_from, $record_per_page);
    } catch (\Exception $e) {
        Log::error($e->getMessage());
        return 'Sorry! Something is wrong with database!';
    }
    $output = '';
    if (empty($res)) {
        $output .= "<div class='jumbotron mx-5'><h1 class='display-4'>Hello,
from Martineez!</h1><p class='lead'>We see you are here for the first time, We give
you some information about us. We are social network only for caterer's. Here we
don't post pic about us, we make some masterpiece of our work, like some cocktail,
dishes or everything else from restaurant, bar, hotel, etc. Take a photo and share
with others here.</p><hr class='my-4'><p>Click on button bellow to share your
picture or some post</p><p class='lead'><button type='button' class='btn btn-
primary' data-toggle='modal' data-target='#exampleModal'>Share Photo</button>
<button type='button' class='btn btn-primary' data-toggle='modal' data-
target='#article-modal'>Share Article</button></p></div>";
    } else {
        foreach ($res as $item) {
            if (empty($item->srcPost)) {
                // TEKSTUALNI POST
                $output .= "<div class='d-flex justify-content-center'><div

```

```

id='post' class='list-group mb-5'><div class='list-group-item list-group-item-
action flex-column align-items-start pb-0'><div class='d-flex w-100 justify-
content-between'>;
    $output .= "<button class='btnDel' rel='".$item->id'><img
src='images/btnDel.png'></button><h6 class='mb-1'>".$item->firstname.' '.$item-
>lastname."</h6><small>".$Carbon\Carbon::parse($item->created_at)->format('d.m.Y
\at H:i')."</small></div><figure class='figure mb-0'><p class='text-
justify'>".$item->title."</p></figure><hr class='my-0'><div class='container py-
1'><div class='row'><button class='ajax-link a-like-com' rel='".$item->id.'"><div
class='container'><div class='row'><i class='material-icons mx-
1'>favorite_border</i>Like</div></div></button><button class='a-like-com commPost'
rel='".$item->id.'"><div class='container'><div class='row'><i class='material-
icons ml-3 mr-1'>folder_open</i>Comment</div></div></button></div></div></div>";
    if($item->numLike!=0){
        $output .= "<div class='list-group-item list-group-item-
action flex-column align-items-start'><a id='".$item->id.'" href=''" class='a-like-
com num viewLikes' rel='".$item->id.'">".$item->numLike." person like</a></div>";}
    else{
        $output .= "<div class='list-group-item list-group-item-
action flex-column align-items-start'><a id='".$item->id.'" href=''" class='a-like-
com'></a></div>";}
    if($item->numComment!=0){
        $output .= "<div id='".$item->id.'"showComm' class='list-
group-item list-group-item-action flex-column align-items-start pb-2'><a href=''"
class='viewComment' rel='".$item->id.'">View Comment</a></div>";}
        $output .= "<div id='".$item->id.'"comm'.'"></div></div></div>";
    }else{
        $output .= "<div class='d-flex justify-content-center'><div
class='list-group mb-5' style='max-width: 542px !important;'><div class='list-
group-item list-group-item-action flex-column align-items-start pb-0'><div
class='d-flex w-100 justify-content-between'><button class='btnDel' rel='".$item-
>id'><img src='images/btnDel.png'></button><h6 class='mb-1'><img src='";
        if($item->srcProfil!=""){
            $output .= $item->srcProfil;
        }
        else{
            $output .= "images/blank-profile.png";
        }
        $output .= "'class='rounded-circle mb-2 mr-2
xs-img'/">".$item->firstname." ".$item-
>lastname."</h6><small>".$Carbon\Carbon::parse($item->created_at)->format('d.m.Y
\at H:i')."</small></div><p>".$item->title."</p><figure class='figure mb-0'><a
href='".$item->srcPost.'" data-lightbox='".$item->id.'"><img src='".$item-
>srcPost.'" class='figure-img img-fluid rounded'></a><figcaption class='figure-
caption'></figcaption></figure><div class='container pb-1'><div class='row'><button
class='ajax-link a-like-com' rel='".$item->id.'"><div class='container'><div
class='row'><i class='material-icons mx-
1'>favorite_border</i>Like</div></div></button><button class='commPost a-like-com'
rel='".$item->id.'"><div class='container'><div class='row'><i class='material-
icons ml-3 mr-1'>folder_open</i>Comment</div></div></button></div></div></div>";
        if($item->numLike!=0){
            $output .= "<div class='list-group-item list-group-item-
action flex-column align-items-start'><a id='".$item->id.'" href=''" class='a-like-
com num viewLikes' rel='".$item->id.'">".$item->numLike." person like</a></div>";}
        else{
            $output .= "<div class='list-group-item list-group-item-
action flex-column align-items-start'><a id='".$item->id.'" href=''" class='a-like-
com'></a></div>";}
        if($item->numComment!=0){
            $output .= "<div id='".$item->id.'"showComm' class='list-
group-item list-group-item-action flex-column align-items-start pb-2'><a href=''"
class='viewComment' rel='".$item->id.'">View Comment</a></div>";}
            $output .= "<div id='".$item->id.'"comm'.'"></div></div></div>";
        }
    }
};

```

```

        try{
            $total_records = $post->getAllForUser();
        }catch (\Exception $e){
            Log::error($e);
            return 'Sorry! Something is wrong with database!';
        }
        $str = $total_records[0]->num;
        $total_pages = ceil($str/$record_per_page);
        $output .= "<div class='pagination pagination-sm justify-content-center d-flex'>";
        for ($i=1; $i<=$total_pages; $i++){
            $output .= "<span class='pagination_link page-link' style='cursor: pointer; border: 1px solid #ccc;' id='\".$i.\"'>\".$i.\"</span>";
            $output .= " ";
        }
        $output .= "</div>";
    }

    return $output;
}
}

```

## ArticleController.php

```

<?php

namespace App\Http\Controllers;

use App\Models\Article;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Log;

class ArticleController extends Controller
{
    public function store(Request $request)
    {
        $rules = [
            'tbArticle' => 'required|regex:/^[a-zA-Z]+(([\',. -][a-zA-Z ])?[a-zA-Z]*)*$/|min:3'
        ];

        $messages = [
            'tbArticle.required' => 'Article cannot be empty!',
            'tbArticle.regex' => 'No special characters please!',
            'tbArticle.min' => 'Minimun three letters!'
        ];

        $request->validate($rules, $messages);
        try{
            $article = new Article();
            $article->setTitle($request->get('tbArticle'));
            $article->setUser_id(session()->get('user')[0]->getId());
            $article->insert();
            return redirect()->back()->with('msg','post successfully added post!');
        }
        catch (\Exception $e){
            Log::error($e);
            return 'Sorry! Something is wrong with database!';
        }
    }
}

```

## CommentController.php



```

<?php

namespace App\Http\Controllers;

use App\Models\Comment;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use Psy\Util\Json;
use Illuminate\Support\Facades\Log;

class CommentController extends Controller
{
    public function store(Request $request)
    {
        $text = $request->get('tbComment');
        $id = $request->get('id');
        $rules=[
            'tbComment' => 'required|min:3',
        ];
        $message = [
            'tbComment' => 'required',
        ];
        $request->validate($rules, $message);
        $comment = new Comment();
        if(session()->has('user')){
            $comment ->setUser_id(session()->get('user')[0]->getId());
        }else{
            $comment->setUser_id(session()->get('admin')[0]->getId());
        }

        $comment ->setPost_id($id);
        $comment ->setText($text);
        try{
            $comment->store();
            return redirect()->back();
        }catch (\Exception $e){
            Log::error($e);
            return 'Sorry! Something is wrong with database!';
        }
    }

    public function show(Request $request){
        $comm = new Comment();
        $comm->setPost_id($request->get('id'));
        $suid = session()->get('user')[0]->getId();
        try{
            $res = $comm->show();
            $output = "";
            foreach ($res as $item){
                $output .="<p class='text-justify nb text-left comment-text font-weight-bold'>$item->text</p>";
                $output .="<p class='text-right nb'><small>$item->firstname $item->lastname";
                if($suid==$item->uid){
                    $output .="<br><button class='btnDelComm' rel='$item->id'><img src='images/rsz_btnDel.png'></button>";
                }
                $output .="</small></p><hr>";
            }
            return Json::encode($output);
        }catch (\Exception $e){
            Log::error($e);
            return 'Sorry! Something is wrong with database!';
        }
    }
}

```

```

    public function delComm(Request $request){
        try{
            $id = $request->get('id');
            DB::table('comment')
                ->where('id','=',$id)
                ->delete();
        }catch (\Exception $e){
            Log::error($e);
            return 'Sorry! Something is wrong with database!';
        }
    }
}

```

## HomeController.php

```

<?php
namespace App\Http\Controllers;

use App\Models\Advertisement;
use App\Models\Image;
use App\Models\Post;
use App\Models\Survey;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Log;

class HomeController extends Controller
{
    private $data;
    private $model;

    public function index()
    {
        $post=new Post();
        $adv = new Advertisement();
        try{
            $this->model=new Survey();
            $id = $this->model->getSurveyShow();
            $this->model->setId($id->survey_id);
            $this->data['survey'] = $this->model->getSurveyById();
            $this->data['post'] = $post->getAllPost();
            $this->data['adv'] = $adv->getAllAdvertisement();
            return view('pages.home',$this->data);
        }catch (\Exception $e){
            Log::error($e);
            return 'Sorry! Something is wrong with database!';
        }
    }
}

```

## ImgController.php

```

<?php
namespace App\Http\Controllers;

use App\Models\Advertisement;
use App\Models\Image;
use App\Models\Post;
use App\Models\Survey;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Log;

```

```

class HomeController extends Controller
{
    private $data;
    private $model;

    public function index()
    {
        $post=new Post();
        $adv = new Advertisement();
        try{
            $this->model=new Survey();
            $id = $this->model->getSurveyShow();
            $this->model->setId($id->survey_id);
            $this->data['survey'] = $this->model->getSurveyById();
            $this->data['post'] = $post->getAllPost();
            $this->data['adv'] = $adv->getAllAdvertisement();
            return view('pages.home',$this->data);
        }catch (\Exception $e){
            Log::error($e);
            return 'Sorry! Something is wrong with database!';
        }
    }
}

```

## IndexController.php

```

<?php
namespace App\Http\Controllers;

use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Log;
use Validator;
use App\Models\User;
use Illuminate\Http\Request;

class IndexController extends Controller
{
    public function index()
    {
        return view('app.index');
    }

    public function showLogin()
    {
        return view('pages.login');
    }

    public function login(Request $request){

        $validator = Validator::make($request->all(), [
            'tbMail' => 'required|email',
            'tbPassword' => 'required|regex:/^(?=.*\d)(?=.*[a-zA-Z])[a-zA-Z0-9]{7,}$/',
        ]);

        if ($validator->fails()) {
            return view('pages.login')
                ->with('msg','Invalid username or password!');
        }
        else{
            try{
                $user = new User();

```

```

$mail = $request->input('tbMail');
$check = $user->checkMail($mail);
if($check!="") {

    $user->setEmail($mail);
    try{
        $hashedPassword = DB::table('user')
            ->where('email','=',$mail)
            ->select('password')
            ->first();
        $hash = $hashedPassword->password;
    }catch (\Exception $e){
        return $e->getMessage();
    }
    $pass = $request->input('tbPassword');
    if(Hash::check($pass, $hash)){
        $res = DB::table('user')
            ->where('email','=',$mail)
            ->first();
        if($res)
        {
            $user->setId($res->id);
            $user->setFirstname($res->firstname);
            $user->setLastname($res->lastname);
            $user->setPosition($res->position);
            $user->setWorkplace($res->workplace);
            $user->setState($res->state);
            $user->setCity($res->city);
            $user->setDateOfBirth($res->dateOfBirth);
            $user->setRole_id($res->role_id);
            if($res->role_id==2){
                $request->session()->push('user',$user);
                return redirect()->route('home');
            }else{
                $request->session()->push('admin',$user);
                return redirect()->route('admin');
            }
        }
    }
    else{
        return view('pages.login')
            ->with('msg','Invalid username or password!');
    }
} else{
    return view('pages.login')
        ->with('msg','Invalid username or password!');
}
} catch (\Exception $e){
    Log::error($e);
    return 'Sorry! Something is wrong with database!';
}
}

}

public function logout(Request $request)
{
    $request->session()->flush();
    return redirect()->route('index');
}

public function store(Request $request){

    //VALIDATION OF DATA

    $request->validate($this->rules(), $this->message());

```

```

        $request->validate([
            'tbMail' => 'required|email',
            'tbPassword' => 'regex:/^(?=.*\d)(?=.*[a-zA-Z])[a-zA-Z0-9]{7,}$/',
        ]);

        //CHECKING EMAIL
        $user = new User();
        $mail = $request->input('tbMail');
        $check = $user->checkMail($mail);
        if($check)
        {
            return redirect()->back()->with('msg','There is already user with this
email address!');
        }
        else
        {
            $user->setFirstname($request->input('tbFirstname'));
            $user->setLastname($request->input('tbLastname'));
            $pass = Hash::make($request->input('tbPassword'));
            $user->setPassword($pass);
            $user->setEmail($request->input('tbMail'));
            $user->setWorkplace($request->input('tbWorkplace'));
            $user->setPosition($request->input('tbPosition'));
            try{
                $id = $user->newUser();
                $user->setId($id);
                $user->setRole_id(2);
                $request->session()->push('user',$user);
                return redirect()->route('profil');
            }catch(\Exception $e){
                Log::error($e);
                return 'Sorry! Something is wrong with database!';
            }
        }
    }
}
}

```

## Profilcontroller.php

```

<?php

namespace App\Http\Controllers;

use App\Models\Image;
use App\Models\Post;
use App\Models\User;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Log;

class ProfilController extends Controller
{
    private $data;

    public function index(Request $request)
    {
        $id = $request->session()->get('user')[0]->getId();
        $user = new User();
        $this->data['user'] = $user->getUserById($id);
        $img = new Image();
        $this->data['profilimg'] = $img->getProfileImg($id);
        $this->data['gallery'] = $img->getGallery($id);

        $post = new Post();
        $post->setUser_id($id);
        try{

```

```

        $this->data['post'] = $post->getAllPostById();
    } catch (\Exception $e) {
        Log::error($e);
        return 'Sorry! Something is wrong with database!';
    }

    return view('pages.profil', $this->data);
}

public function edit(Request $request)
{
    try {
        $request->validate($this->rules(), $this->message());
        $user = new User();
        $user->setFirstname($request->input('tbFirstname'));
        $user->setLastname($request->input('tbLastname'));
        $user->setWorkplace($request->input('tbWorkplace'));
        $user->setPosition($request->input('tbPosition'));
        $user->setDateOfBirth($request->input('tbDate'));
        $id = $request->session()->get('user')[0]->getId();

        if ($request->input('tbCity') == "" && $request->input('tbState') == "")
        {
            $user->editUser($id);
        }
        else
        {
            $rules = [
                'tbCity' => 'regex:/^[a-zA-Z]+(([\',. -][a-zA-Z ])?[a-zA-Z]*)*$/ ',
                'tbState' => 'regex:/^[a-zA-Z]+(([\',. -][a-zA-Z ])?[a-zA-Z]*)*$/ ',
            ];
            $message = [
                'tbCity.regex' => 'Please enter both, State and City!',
                'tbState.regex' => 'Please enter both, State and City!'
            ];
            $request->validate($rules, $message);
            $user->setCity($request->input('tbCity'));
            $user->setState($request->input('tbState'));
            $user->editUser($id);
        }
        return redirect()->back()->with('msg', 'You have successfully changed the data');
    } catch (\Exception $e) {
        Log::error($e);
        return 'Sorry! Something is wrong with database!';
    }
}
}

```

## CheckAdmin.php

```

<?php

namespace App\Http\Middleware;

use Closure;

class CheckAdmin
{
    /**
     * Handle an incoming request.
     *
     * @param \Illuminate\Http\Request $request
     * @param \Closure $next
     * @return mixed
     */
}

```

```

    */
    public function handle($request, Closure $next)
    {
        if (!$request->session()->has('admin')) {
            return abort(404);
        }
        return $next($request);
    }
}

```

## checkSession.php

```

<?php

namespace App\Http\Middleware;

use Closure;

class CheckSession
{
    /**
     * Handle an incoming request.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  \Closure  $next
     * @return mixed
     */
    public function handle($request, Closure $next)
    {
        if (!$request->session()->has('user')) {
            return abort(404);
        }
        return $next($request);
    }
}

```

## Advertisement.php

```

<?php
/**
 * Created by PhpStorm.
 * User: Laba
 * Date: 3/14/2018
 * Time: 12:58 AM
 */

namespace App\Models;

use Illuminate\Support\Facades\DB;

class Advertisement
{
    //region data
    private $id;
    private $title;
    private $alt;
    private $src;
    private $text;
    //endregion

    //region getter and setter
    public function getId() {
        return $this->id;
    }
}

```

```

public function setId($id) {
    $this->id = $id;
}

public function getTitle() {
    return $this->title;
}

public function setTitle($title) {
    $this->title = $title;
}

public function getAlt() {
    return $this->alt;
}

public function setAlt($alt) {
    $this->alt = $alt;
}

public function getSrc() {
    return $this->src;
}

public function setSrc($src) {
    $this->src = $src;
}

public function getText() {
    return $this->text;
}

public function setText($text) {
    $this->text = $text;
}
//endregion

public function getAllAdvertisement() {
    return DB::table('advertisement')
        ->get();
}

public function insertAdv() {
    DB::table('advertisement')->insert(
        [
            'text' => $this->text,
            'src' => $this->src,
            'alt' => $this->src,
            'title' => $this->title
        ]
    );
}

public function getAdvById() {
    return DB::table('advertisement')
        ->where('id', '=', $this->id)
        ->get();
}

public function updateAdv() {
    DB::table('advertisement')
        ->where('id', $this->id)
        ->update([
            'text' => $this->text,
            'src' => $this->src,
            'alt' => $this->alt,
            'title' => $this->title
        ]);
}

```



```

    }

    public function delete() {
        DB::table('advertisement')
            ->where('id','=', $this->id)
            ->delete();
    }
}

```

## Article.php

```

<?php
/**
 * Created by PhpStorm.
 * User: Laba
 * Date: 2/28/2018
 * Time: 9:37 PM
 */

namespace App\Models;

use Illuminate\Support\Facades\DB;

class Article
{
    //region data
    private $id;
    private $title;
    private $user_id;
    private $created_at;
    //endregion
    //region getter and setter
    public function getId() {
        return $this->id;
    }

    public function setId($id) {
        $this->id = $id;
    }

    public function getTitle() {
        return $this->title;
    }

    public function setTitle($title) {
        $this->title = $title;
    }

    public function getUser_id() {
        return $this->user_id;
    }

    public function setUser_id($user_id) {
        $this->user_id = $user_id;
    }

    public function getCreated_at() {
        return $this->created_at;
    }

    public function setCreated_at($created_at) {
        $this->created_at = $created_at;
    }
    //endregion
}

```

```

    public function insert() {
        DB::table('post')
            ->insert([
                'title' => $this->title,
                'user_id' => $this->user_id,
            ]);
    }
}

```

## Comment.php

```

<?php
/**
 * Created by PhpStorm.
 * User: Laba
 * Date: 3/4/2018
 * Time: 6:27 PM
 */

namespace App\Models;

use Illuminate\Support\Facades\DB;

class Comment
{
    //region data
    private $id;
    private $user_id;
    private $post_id;
    private $text;
    private $created_at;
    //endregion

    //region getter and setter
    public function getId() {
        return $this->id;
    }

    public function setId($id) {
        $this->id = $id;
    }

    public function getUser_id() {
        return $this->user_id;
    }

    public function setUser_id($user_id) {
        $this->user_id = $user_id;
    }

    public function getPost_id() {
        return $this->post_id;
    }

    public function setPost_id($post_id) {
        $this->post_id = $post_id;
    }

    public function getText() {
        return $this->text;
    }
}

```

```

    public function setText($text){
        $this->text = $text;
    }

    public function getCreated_at(){
        return $this->created_at;
    }

    public function setCreated_at($created_at){
        $this->created_at = $created_at;
    }
    //endregion

    public function store()
    {
        DB::table('comment')
            ->insert([
                'user_id' => $this->user_id,
                'post_id' => $this->post_id,
                'text' => $this->text,
            ]);
    }

    public function show()
    {
        return DB::table('comment')
            ->join('user', 'user.id', '=', 'comment.user_id')
            ->where('post_id', '=', $this->post_id)
            ->select('user.firstname', 'user.lastname', 'comment.text',
'comment.created_at', 'user.id as uid', 'comment.id')
            ->get();
    }
}

```

## Image.php

```

<?php
/**
 * Created by PhpStorm.
 * User: Laba
 * Date: 2/23/2018
 * Time: 2:45 PM
 */

namespace App\Models;

use Illuminate\Support\Facades\DB;

class Image
{
    //region data
    private $id;
    private $src;
    private $alt;
    private $active;
    private $user_id;
    //endregion

    //region getter and setter
    public function getId(){
        return $this->id;
    }

    public function setId($id){
        $this->id = $id;
    }
}

```

```

    }

    public function getUserId() {
        return $this->id;
    }

    public function setUserId($id) {
        $this->id = $id;
    }

    public function getAlt()
    {
        return $this->alt;
    }

    public function setAlt($alt) {
        $this->alt = $alt;
    }

    public function getSrc() {
        return $this->src;
    }

    public function setSrc($src) {
        $this->src = $src;
    }

    public function getActive() {
        return $this->active;
    }

    public function setActive($active) {
        $this->active = $active;
    }
    //endregion

    public function insertImg()
    {
        return DB::table('image')
            ->insertGetId([
                'src' => $this->getSrc(),
                'active' => $this->getActive(),
                'alt' => $this->getAlt(),
                'user_id' => $this->getUserId()
            ]);
    }

    public function getProfileImg($id) {

        return DB::table('image')
            ->join('profileimg', 'image.id', '=', 'profileimg.img_id')
            ->where('profileimg.user_id', '=', $id)
            ->select('image.*')
            ->orderByRaw('profileimg.created_at DESC')
            ->first();
    }

    public function getGallery($id) {

        return DB::table('image')
            ->join('gallery', 'image.id', '=', 'gallery.img_id')
            ->where('gallery.user_id', '=', $id)
            ->select('image.*')
            ->orderByRaw('gallery.created_at DESC')
            ->get();
    }
}

```

## Post.php

```
<?php
/**
 * Created by PhpStorm.
 * User: Laba
 * Date: 2/27/2018
 * Time: 7:01 PM
 */

namespace App\Models;

use Illuminate\Support\Facades\DB;

class Post
{
    //region data
    private $id;
    private $title;
    private $user_id;
    private $img_id;
    private $created_at;
    //endregion

    //region getter and setter
    public function getId(){
        return $this->id;
    }

    public function setId($id){
        $this->id = $id;
    }

    public function getTitle(){
        return $this->title;
    }

    public function setTitle($title){
        $this->title = $title;
    }

    public function getUser_id(){
        return $this->user_id;
    }

    public function setUser_id($user_id){
        $this->user_id = $user_id;
    }

    public function getImg_id(){
        return $this->img_id;
    }

    public function setImg_id($img_id){
        $this->img_id = $img_id;
    }

    public function getCreated_at(){
        return $this->created_at;
    }

    public function setCreated_at($created_at){
        $this->created_at = $created_at;
    }
    //endregion

    public function insertPost()
```

```

{
    DB::table('post')
        ->insert([
            'title' => $this->title,
            'user_id' => $this->user_id,
            'img_id' => $this->img_id,
        ]);
}

public function getAllPost()
{
    return DB::select('SELECT p.title, p.id, p.created_at, i.src as srcProfil,
i.alt as altProfil, i2.src as srcPost, i2.alt as altPost , u.firstname,
u.lastname,u.id as uid, COUNT(DISTINCT v.id) as numLike, COUNT(c.id) as numComment
FROM post p
LEFT JOIN vote v
ON v.post_id = p.id
LEFT JOIN comment c ON c.post_id = p.id
LEFT JOIN profileimg pi
ON p.user_id = pi.user_id
LEFT JOIN image i ON pi.img_id =i.id
LEFT JOIN image i2 on p.img_id = i2.id
JOIN user u ON u.id = p.user_id
WHERE i2.active = 1 OR i2.active is null
GROUP BY p.id
ORDER BY p.created_at DESC');
}

public function getAllPostById()
{
    return DB::select('SELECT p.title, p.id, p.created_at, i.src as srcProfil,
i.alt as altProfil, i2.src as srcPost, i2.alt as altPost , u.firstname,
u.lastname,u.id as uid, COUNT(DISTINCT v.id) as numLike, COUNT(DISTINCT c.id) as
numComment
FROM post p
LEFT JOIN vote v
ON v.post_id = p.id
LEFT JOIN comment c ON c.post_id = p.id
LEFT JOIN profileimg pi
ON p.user_id = pi.user_id
LEFT JOIN image i ON pi.img_id =i.id
LEFT JOIN image i2 on p.img_id = i2.id
JOIN user u ON u.id = p.user_id
WHERE (i2.active = 1 OR i2.active is null) AND p.user_id = ?
GROUP BY p.id
ORDER BY p.created_at DESC', [$this->user_id]);
}

public function getAllPostForPagination($start, $rec){
    return DB::select('SELECT p.title, p.id, p.created_at, i.src as srcProfil,
i.alt as altProfil, i2.src as srcPost, i2.alt as altPost , u.firstname, u.lastname,
COUNT(DISTINCT v.id) as numLike, COUNT(DISTINCT c.id) as numComment
FROM post p
LEFT JOIN vote v
ON v.post_id = p.id
LEFT JOIN comment c ON c.post_id = p.id
LEFT JOIN profileimg pi
ON p.user_id = pi.user_id
LEFT JOIN image i ON pi.img_id =i.id
LEFT JOIN image i2 on p.img_id = i2.id
JOIN user u ON u.id = p.user_id
WHERE (i2.active = 1 OR i2.active is null) AND p.user_id = ?
GROUP BY p.id
ORDER BY p.created_at DESC LIMIT ?, ?', [session()->get('user')[0]-
>getId(), $start, $rec]);
}

public function getAllForUser(){
    return DB::select('SELECT Count(id) as num FROM post WHERE user_id =
?', [session()->get('user')[0]->getId()]);
}

```

```

    }
}

```

## Survey.php

```

<?php
/**
 * Created by PhpStorm.
 * User: Laba
 * Date: 3/5/2018
 * Time: 1:31 PM
 */

namespace App\Models;

use Illuminate\Support\Facades\DB;

class Survey
{
    //region data
    private $id;
    private $user_id;
    private $question;
    //endregion

    //region getter and setter
    public function getId(){
        return $this->id;
    }

    public function setId($id){
        $this->id = $id;
    }

    public function getUser_id(){
        return $this->user_id;
    }

    public function setUser_id($user_id){
        $this->user_id = $user_id;
    }

    public function getQuestion(){
        return $this->question;
    }

    public function setQuestion($question){
        $this->question = $question;
    }
    //endregion

    public function getSurveyById()
    {
        return DB::table('survey-question')
            ->join('question','survey-question.question_id','=','question.id')
            ->where('survey-question.survey_id','=', $this->id)
            ->get();
    }

    public function getStatistic($id){
        return DB::select('SELECT q.question, q.id, AVG(answer) as avg FROM
question q JOIN `survey-question` sq ON q.id = sq.question_id JOIN answer a ON
a.question_id = q.id WHERE sq.survey_id = ? GROUP BY a.question_id',[$id]);
    }
}

```

```

    public function getNumUser($id, $answer){
        return DB::select('SELECT COUNT(user_id) as num FROM answer WHERE
question_id = ? AND answer = ?',[$id, $answer]);
    }

    public function updateQuestion($text, $id){
        DB::table('question')
            ->where('id', $id)
            ->update(['question' => $text]);
    }

    public function insertSurvey($name){
        return DB::table('survey')
            ->insertGetId([
                'name' => $name,
            ]);
    }

    public function insertQuestion($q){
        return DB::table('question')
            ->insertGetId([
                'question' => $q,
            ]);
    }

    public function insertQuestionSurvey($id, $q1, $q2, $q3){
        return DB::table('survey-question')
            ->insert([
                ['survey_id' => $id,
                'question_id' => $q1,
                ],[
                'survey_id' => $id,
                'question_id' => $q2,
                ],[
                'survey_id' => $id,
                'question_id' => $q3,
                ]
            ]);
    }

    public function deleteSurvey($id){
        DB::table('survey')
            ->where('id', '=', $id)
            ->delete();
    }

    public function getSurveyShow(){
        return DB::table('survey-show')
            ->where('id', '=', 1)
            ->select('survey_id')
            ->first();
    }

    public function setSurveyActive($id){
        DB::table('survey-show')
            ->where('id', '=', 1)
            ->update(['survey_id'=> $id]);
    }
}

```

User.php

```
<?php
```



```

namespace App\Models;

use Illuminate\Support\Facades\DB;

class User
{
    //region data
    private $id;
    private $firstname;
    private $lastname;
    private $password;
    private $email;
    private $workplace;
    private $position;
    private $city;
    private $state;
    private $dateOfBirth;
    private $role_id;
    //endregion

    //region getter and setter
    public function getId(){
        return $this->id;
    }

    public function setId($id){
        $this->id = $id;
    }

    public function getFirstname(){
        return $this->firstname;
    }

    public function getRole_id(){
        return $this->role_id;
    }

    public function setRole_id($roleId){
        $this->role_id = $roleId;
    }

    public function setFirstname($firstname){
        $this->firstname = $firstname;
    }

    public function getLastname(){
        return $this->lastname;
    }

    public function setLastname($lastname){
        $this->lastname = $lastname;
    }

    public function getPassword(){
        return $this->password;
    }

    public function setPassword($password){
        $this->password = $password;
    }

    public function getEmail(){
        return $this->email;
    }

    public function setEmail($email){
        $this->email = $email;
    }
}

```

```

    }

    public function getWorkplace() {
        return $this->workplace;
    }

    public function setWorkplace($workplace) {
        $this->workplace = $workplace;
    }

    public function getPosition() {
        return $this->position;
    }

    public function setPosition($position) {
        $this->position = $position;
    }

    public function getCity() {
        return $this->city;
    }

    public function setCity($city) {
        $this->city = $city;
    }

    public function getState() {
        return $this->state;
    }

    public function setState($state) {
        $this->state = $state;
    }

    public function getDateOfBirth() {
        return $this->dateOfBirth;
    }

    public function setDateOfBirth($dateOfBirth) {
        $this->dateOfBirth = $dateOfBirth;
    }
//endregion

    public function checkMail($mail)
    {
        return DB::table('user')
            ->where('email', $mail)
            ->first();
    }

    public function getAllUsers() {
        return DB::table('user')
            ->where('role_id', '=', 2)
            ->get();
    }

    public function newUser()
    {
        return DB::table('user')->insertGetId(
            [ 'firstname' => $this->firstname,
              'lastname' => $this->lastname,
              'password' => $this->password,
              'email' => $this->email,
              'workplace' => $this->workplace,
              'position' => $this->position,
              'role_id' => 2,
            ]
        );
    }

```

```

    }

    public function login(User $user)
    {
        return DB::table('user')
            ->where('password', '=', $user->getPassword())
            ->where('email', '=', $user->getEmail())
            ->first();
    }

    public function getUserById($id){
        return DB::table('user')
            ->where('id', $id)
            ->first();
    }

    public function editUser($id)
    {
        if($this->state=="" && $this->city=="") {
            DB::table('user')
                ->where('id', $id)
                ->update([
                    'firstname' => $this->firstname,
                    'lastname' => $this->lastname,
                    'workplace' => $this->workplace,
                    'position' => $this->position
                ]);
        }
        else if($this->dateOfBirth==""){
            DB::table('user')
                ->where('id', $id)
                ->update([
                    'firstname' => $this->firstname,
                    'lastname' => $this->lastname,
                    'workplace' => $this->workplace,
                    'position' => $this->position,
                    'city' => $this->city,
                    'state' => $this->state,
                ]);
        }
        else{
            DB::table('user')
                ->where('id', $id)
                ->update([
                    'firstname' => $this->firstname,
                    'lastname' => $this->lastname,
                    'workplace' => $this->workplace,
                    'position' => $this->position,
                    'city' => $this->city,
                    'state' => $this->state,
                    'dateOfBirth' => $this->dateOfBirth
                ]);
        }
    }

    public function regUser() {
        DB::table('user')
            ->insert([
                'firstname' => $this->firstname,
                'lastname' => $this->lastname,
                'workplace' => $this->workplace,
                'position' => $this->position,
                'city' => $this->city,
                'state' => $this->state,
                'dateOfBirth' => $this->dateOfBirth,
                'password' => $this->password,
                'email' => $this->email,
                'role_id' => 2,
            ]);
    }
}

```

```
}
```

## Vote.php

```
<?php
/**
 * Created by PhpStorm.
 * User: Laba
 * Date: 3/2/2018
 * Time: 12:34 AM
 */

namespace App\Models;

use Illuminate\Support\Facades\DB;

class Vote
{
    //region data
    private $id;
    private $post_id;
    private $user_id;
    //endregion

    //region getter and setter
    public function getId(){
        return $this->id;
    }

    public function setId($id){
        $this->id = $id;
    }

    public function getPost_id(){
        return $this->post_id;
    }

    public function setPost_id($post_id){
        $this->post_id = $post_id;
    }

    public function getUser_id(){
        return $this->user_id;
    }

    public function setUser_id($user_id){
        $this->user_id = $user_id;
    }
    //endregion

    public function insertLike()
    {
        DB::table('vote')
            ->insert([
                'post_id' => $this->post_id,
                'user_id' => $this->user_id,
            ]);
    }

    public function checkIfLiked()
    {
        return DB::table('vote')->where([
            ['post_id', '=', $this->post_id],
            ['user_id', '=', $this->user_id],
        ])->first();
    }
}
```

```

    }

    public function unlike()
    {
        DB::table('vote')->where([
            ['post_id', '=', $this->post_id],
            ['user_id', '=', $this->user_id],
        ])->delete();
    }

    public function showNumLikes()
    {
        return DB::table('vote')
            ->select(DB::raw('count(post_id) as likes'))
            ->where('post_id', '=', $this->post_id)
            ->groupBy('post_id')
            ->get();
    }

    public function likers()
    {
        return DB::table('user')
            ->select('user.firstname', 'user.lastname')
            ->join('vote', 'vote.user_id', '=', 'user.id')
            ->where('vote.post_id', '=', $this->post_id)
            ->get();
    }
}

```

## Admin-layout.blade.php

```

<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">

    <title>Admin Panel</title>

    <!-- Bootstrap core CSS -->
    <link href="{{asset('css/bootstrap.min.css')}}" rel="stylesheet">
    <script src="{{asset('js/jquery-3.3.1.min.js')}}"></script>
</head>

<body>
<nav class="navbar navbar-dark sticky-top bg-dark flex-md-nowrap p-0">
    <a class="navbar-brand col-sm-3 col-md-2 mr-0"
href="{{url('/').'/admin'}}">Martinez</a>
    <ul class="navbar-nav px-3">
        <li class="nav-item text-nowrap">
            <a class="nav-link" href="{{url('/').'/admin/logout'}}">Sign out</a>
        </li>
    </ul>
</nav>

<div class="container-fluid">
    <div class="row">
        <nav class="col-md-2 bg-light sidebar">
            <div class="sidebar-sticky">
                <ul class="nav flex-column">
                    <li class="nav-item">
                        <a class="nav-link active" href="{{url('/').'/admin'}}">
                            <span data-feather="users"></span>

```

```

        Users <span class="sr-only"></span>
    </a>
</li>
<li class="nav-item">
    <a class="nav-link" href="{{url('/')."/admin/post"}}">
        <span data-feather="file"></span>
        Posts
    </a>
</li>
<li class="nav-item">
    <a class="nav-link" href="{{url('/')."/admin/comment"}}">
        <span data-feather="message-square"></span>
        Comments
    </a>
</li>
<li class="nav-item">
    <a class="nav-link"
href="{{url('/')."/admin/activities"}}">
        <span data-feather="activity"></span>
        Activities
    </a>
</li>
<li class="nav-item">
    <a class="nav-link" href="{{url('/')."/admin/survey"}}">
        <span data-feather="send"></span>
        Survey
    </a>
</li>
<li class="nav-item">
    <a class="nav-link" href="{{url('/')."/admin/adv"}}">
        <span data-feather="tv"></span>
        Advertisement
    </a>
</li>
</ul>
</div>
</nav>
<main role="main" class="col-md-9 ml-sm-auto col-lg-10 pt-3">
    @yield('content')
</main>
</div>
</div>
</div>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpilMquVdAyjUar5+76PVCmYl"
crossorigin="anonymous"></script>

<script src="{{asset('js/admin.js')}}"></script>
<script src="{{asset('js/jquery-validation.js')}}"></script>
<script type="text/javascript" src="//cdnjs.cloudflare.com/ajax/libs/jquery-
validate/1.13.1/additional-methods.js"></script>

<!-- Icons -->
<script src="https://unpkg.com/feather-icons/dist/feather.min.js"></script>
<script>
    feather.replace()
</script>
</body>
</html>

```

## Index.blade.php

```

<!doctype html>
<html lang="en">
<head>
    <!-- Required meta tags -->

```

```

    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons"
rel="stylesheet">
    <link href="https://fonts.googleapis.com/css?family=Berkshire+Swash"
rel="stylesheet">
    <link rel="stylesheet" href="{{asset('css/bootstrap.min.css')}}">
    <link href="{{asset('css/style.css')}}" rel="stylesheet">
    <title>Martinez</title>

</head>
<body>
<nav class="navbar navbar-expand-lg navbar-light bg-light">
    <div class="container">
        <a class="navbar-brand logo-nav" href="{{url('/') }}">Martinez</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarTogglerDemo02" aria-controls="navbarTogglerDemo02" aria-
expanded="false" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarTogglerDemo02">
            <span class="navbar-nav mr-auto mt-2 mt-lg-0">
                </span>

                {{--FORMA LOGIN--}}
                <form class="form-inline my-2 my-lg-0" id="loginForm"
action="{{url('/')}. "/login"}}" method="post">
                    {{csrf_field()}}
                    <label for="tbUsername"></label>
                    <input class="form-control mr-sm-2 mt-2 mt-lg-0" type="text"
placeholder="Username" name="tbMail" id="tbMail">
                    <label for="tbPassword"></label>
                    <input class="form-control mr-sm-2 mt-2 mt-lg-0" type="text"
placeholder="Password" name="tbPassword" id="tbPassword">
                    <button class="btn btn-primary mt-2 mt-lg-0"
type="submit">Login</button>
                </form>
                {{--END LOGIN FORM--}}

            </div>
        </div>
    </nav>

    <div class="container">
        <div class="col-md-6 offset-md-3">
            <!-- form card register -->
            <div class="card card-outline-secondary mt-5">
                <div class="card-header">
                    <h3 id="logo">Get Started it's free</h3>
                    <p>Make your account</p>
                </div>

                <div class="card-block">
                    <form class="form register-form" id="register-form"
action="{{url('/')}. "/registerUser"}}" method="post">
                        {{csrf_field()}}
                        <div class="form-group">
                            <div class="input-group mb-3">
                                <div class="input-group-prepend">
                                    <span class="input-group-text" id="basic-addon1"><i
class="material-icons">account_circle</i></span>
                                </div>
                                <input type="text" name="tbFirstname" class="form-
control" placeholder="Firstname">
                            </div>
                            <div id="firstname"></div>
                        </div>
                    </form>
                </div>
            </div>
        </div>
    </div>

```

```

        <div class="form-group">
            <div class="input-group mb-3">
                <div class="input-group-prepend">
                    <span class="input-group-text" id="basic-addon1"><i
class="material-icons">account_circle</i></span>
                </div>
                <input type="text" class="form-control"
placeholder="Lastname" name="tbLastname">
            </div>
            <div id="lastname"></div>
        </div>

        <div class="form-group">
            <div class="input-group mb-3">
                <div class="input-group-prepend">
                    <span class="input-group-text" id="basic-addon1"><i
class="material-icons">email</i></span>
                </div>
                <label for="tbMail"></label>
                <input type="email" class="form-control"
placeholder="Email" name="tbMail">
            </div>
            <div id="mail"></div>
            @if (session('msg'))
                <div class="alert alert-danger">
                    {{ session('msg') }}
                </div>
            @endif
        </div>

        <div class="form-group">
            <div class="input-group mb-3">
                <div class="input-group-prepend">
                    <span class="input-group-text" id="basic-addon1"><i
class="material-icons">lock</i></span>
                </div>
                <label for="tbPassword"></label>
                <input type="password" class="form-control"
placeholder="Password" name="tbPassword">
            </div>
            <div id="password"></div>
        </div>

        <div class="form-group">
            <div class="input-group mb-3">
                <div class="input-group-prepend">
                    <span class="input-group-text" id="basic-addon1"><i
class="material-icons">restaurant_menu</i></span>
                </div>
                <label for="tbWorkplace"></label>
                <input type="text" class="form-control"
placeholder="Workplace" name="tbWorkplace">
            </div>
            <div id="workplace"></div>
        </div>

        <div class="form-group">
            <div class="input-group mb-3">
                <div class="input-group-prepend">
                    <span class="input-group-text" id="basic-addon1"><i
class="material-icons">local_bar</i></span>
                </div>
                <label for="tbPosition"></label>
                <input type="text" class="form-control"
placeholder="Position" name="tbPosition">
            </div>
            <div id="position"></div>
        </div>

        <div class="form-group">
            <button type="submit" class="btn btn-primary btn-block"
name="btnRegister">Register</button>
        </div>

```



```

        </form>
        <p class="ml-3">Already have an account?<a
href="{{url('/').'/showLog'}}"> Sign in here!</a></p>
    </div>
    @if ($errors->any())
        <div class="alert alert-danger">
            <ul>
                @foreach ($errors->all() as $error)
                    <li>{{ $error }}</li>
                @endforeach
            </ul>
        </div>
    @endif
</div>
<!-- /form card register -->
</div>
</div>
@include('components.footer')
<script
    src="http://code.jquery.com/jquery-3.1.1.min.js"
    integrity="sha256-hVVnYaiADRTO2PzUGmuLJr8BLUSjGIZsDYGmIJLv2b8="
    crossorigin="anonymous"></script>
<script src="{{asset('js/jquery-validation.js')}}"></script>
<script src="{{asset('js/regform-validate.js')}}"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"
crossorigin="anonymous"></script>
</body>
</html>

```

## Layout.blade.php

```

<!doctype html>
<html lang="en">
<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons"
rel="stylesheet">
    <link href="https://fonts.googleapis.com/css?family=Berkshire+Swash"
rel="stylesheet">
    <title>Martinez</title>
    <link rel="stylesheet" href="{{asset('css/bootstrap.min.css')}}">
    <link href="{{asset('css/style.css')}}" rel="stylesheet">
    <link href="{{asset('css/lightbox.css')}}" rel="stylesheet">
    <script src="{{asset('js/jquery-3.3.1.min.js')}}"></script>
</head>
<body>

{{--NAVIGATION--}}
<nav id="mainNav" class="nav navbar navbar-expand-lg navbar-light bg-light div-
sticky">
    <div class="container">
        <a class="navbar-brand logo-nav" href="{{url('/').'/home'}}"
style="color:white">Martinez</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarTogglerDemo02" aria-controls="navbarTogglerDemo02" aria-
expanded="false" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarTogglerDemo02">

```

```

        {{--SEARCH FORM--}}
        <form class="input-group mr-5 mt-2 mt-lg-0" id="search-form"
autocomplete="off">
            {{csrf_field()}}
            <input type="text" class="form-control" placeholder="Search for
people..." name="tbSearch" id="myInput">
            <div class="input-group-append">
                <span class="input-group-text searchTxt"><button type="submit"
class="btn btn-light" id="btnSearch">Search</button></span>
            </div>
        </form>
        <ul id="mainUl" class="navbar-nav my-2 my-lg-0 ml-5">
            <li class="nav-item mx-1">
                <a class="nav-link btn btn-primary"
href="{{url('/',) . '/home'}}">Home</a>
            </li>
            <li id="mainLi" class="nav-item mx-1">
                <a class="nav-link btn btn-primary"
href="{{url('/',) . '/profil'}}">
                    {{session()->get('user')[0]->getFirstname()}}
                </a>
            </li>
            <li class="nav-item mx-1">
                <a class="nav-link btn btn-primary"
href="{{url('/',) . '/logout'}}">Logout</a>
            </li>
        </ul>
    </div>
</div>
</nav>
@if ($errors->any())
    <div class="alert alert-danger text-center">
        <ul>
            @foreach ($errors->all() as $error)
                <li>{{ $error }}</li>
            @endforeach
        </ul>
    </div>
@endif
{{--END NAVIGATION--}}

{{--MODAL FOR POST--}}
@include('components.modal')
@include('components.article-modal')
{{--CONTENT--}}
<div class="mt-1">
    @yield('content')
</div>

{{--FOOTER--}}
@include('components.footer')
<a href="#" id="back-to-top" class="back-to-top" title="Back to top"></a>
<script src="{{asset('js/auto-com.js')}}"></script>
<script src="{{asset('js/lightbox-plus-jquery.min.js')}}"></script>
<script src="{{asset('js/script.js')}}"></script>
<script src="{{asset('js/jquery-validation.js')}}"></script>
<script type="text/javascript" src="//cdnjs.cloudflare.com/ajax/libs/jquery-
validate/1.13.1/additional-methods.js"></script>
<script src="{{asset('js/regform-validate.js')}}"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"
crossorigin="anonymous"></script>
</body>
</html>

```