

## SOMETHING TO UNDERSTAND BEFORE HAND:

1. IndexController.php lying inside app/http/controller contains functions which returns view such as Home, Request New account for Employee or Guest , Password Change, Reset password using Email and Status.
2. AdminController.php lying in app/http/controllers contains all functions which are performed by Admin such as Manage Users, LogViewer, Deadline.
3. EmployeeController.php lying in app/http/controllers contains all functions which are related with employee information.
4. HomeController.php lying in app/http/controllers contains all functions which are related with non-admin user, such as change password, search users, edit profile etc.
5. GuestController.php lying in app/http/controller contains all functions which are related with guest information.
6. adldap.php lying in config contains ldap connection configuration.
7. LoginController.php lying in app/http/controller/auth contains login functionality.
8. HomeController.php lying in app/http/controller contains all functionalities which are related with non-admin.
9. LogViewerController.php lying in vendor/rap2hpoutre/laravel-log-viewer/src/controller contains index() function which returns contains of log file (located at storage/logs/laraval.log)
10. main.blade.php ( index page) lying in resources/views/layouts is the file executed first.
11. index.blade.php lying in resources/views/index/Admin is the file(master page) executed first at admin side.
12. display\_main.blade.php lying in resources/views/layouts is the file(master page) executed first at non-admin side.

## PROJECT DIRECTORY STRUCTURE:

- Web.php :- contains following routes of web portal :

1. `Route::get('/', function () {return view('auth.login'); });`

`// returns default page for login`

2. `Route::get('/refreshcaptcha',function (){`

`return captcha_img('flat');`

`})->name('refreshCaptcha');`

`// returns refresh captcha.`

3. Routes for IndexController

### A. STATUS Routes

1. `Route::get('/index/statusView','IndexController@getStatusView')->name('getStatusView'); // executes getStatusView() in IndexController.`

2. `Route::post('/index/statusInfo','IndexController@getStatus')->name('getStatus'); // executes getStatus() in IndexController.`

### B. CHANGE PASSWORD Routes

3. `Route::get('/index/passwordchangeView','IndexController@getPasswordChangeView')->name('getPasswordChangeView'); // executes getPasswordChangeView() in IndexController.`

4. `Route::post('/index/passwordchange','IndexController@passwordChange')->name('IndexPasswordChange'); // executes passwordChange() in IndexController.`

### C. RESET PASSWORD USING EMAIL Routes

5. `Route::get('/index/passwordResetEmailView','IndexController@getPasswordResetEmail')->name('passwordResetEmail'); // executes getPasswordResetEmail() in IndexController.`

6. Route::post('/index/SentResetPasswordLink', 'IndexController@SentResetPasswordLink')->name('SentResetPasswordLink'); // executes SentResetPasswordLink() in IndexController.

7. Route::get('/index/{username}/PasswordResetEmailView', 'IndexController@PasswordResetEmailView')->name('PasswordResetEmail'); // executes PasswordResetEmailView() in IndexController.

8. Route::post('/index/PasswordResetEmail/{username}', 'IndexController@changePasswordEmail')->name('changePasswordEmail'); // executes changePasswordEmail() in IndexController.

#### D. CREATE EMPLOYEE and GUEST ROUTES

9. Route::resource('/home', 'HomeController'); // routes for all function available in Employee Controller.

10. Route::post('/index/check\_username\_unique', 'IndexController@checkUsername')->name('checkusername'); // executes checkUsername() in

11. Route::post('/index/divisionHead', 'IndexController@getDivisionHead')->name('getDivHead'); // executes getDivisionHead() in IndexController.

12. Route::post('/index/getDept', 'IndexController@getDepartmentOfOwner')->name('getDepartmentOfOwner'); // executes getDepartmentOfOwner() in IndexController. // executes getDepartmentOfOwner() in IndexController.

#### E. Non-Admin Home Page (Employee and Home Controller)

13. Route::resource('/home', 'HomeController'); // all routes for HomeController

14. Route::post('/employee/upload', 'HomeController@upload')->name('upload'); // executes upload() in HomeController.

15. Route::get('/employee/jpegphoto/{id}', 'HomeController@photo')->name('jpegphoto'); // executes photo() in HomeController.

16. Route::post('/employee/search', 'HomeController@search')->name('search'); // executes search() in HomeController.

17. Route::get('/employee/logout','HomeController@logout')->name('logout');  
// executes logout() in HomeController.

### E.1 Change Password Non-Admin

18. Route::get('/employee/  
password\_change\_view','HomeController@password\_change\_view')-  
>name('password\_view'); //executes password\_view() in HomeController.

19. Route::post('/employee/  
passwordChange','EmployeeController@changePassword')-  
>name('changepassword'); //executes changePassword() in EmployeeController.

### E.2 Employee profile Update

20. Route::get('/employee/profile/{id}','HomeController@profile')-  
>name('profile'); // executes profile() in HomeController.

21. Route::post('/employee/profileUpdate','EmployeeController@update')-  
>name('profileupdate'); // executes update() in profileupdate.

### F. ADMIN routes

#### // ADMIN CONTROLLER

22. Route::get('/Admin/home','AdminController@index')-  
>name('Admin.index'); // executes index() in AdminController.

23. Route::get('/Admin/allUsers','AdminController@getAllUsers')-  
>name('getAllUsers'); // executes getAllUsers() in AdminController.

24. Route::post('/Admin/createUser','AdminController@createUser')-  
>name('createUser'); //executes createUser() in AdminController.

25. Route::get('Admin/deleteUser/{id}','AdminController@deleteUser')-  
>name('deleteUser'); // executes deleteUser() in AdminController.

26. Route::get('/Admin/lockedUsers','AdminController@getLockedUsers')-  
>name('getLockedUsers'); // executes getLockedUsers() in AdminController.

27. Route::get('/Admin/  
unlockedUsers','AdminController@getUnLockedUsers')-  
>name('getUnLockedUsers'); // executes getUnLockedUsers() in AdminController.

28. `Route::get('/Admin/guests','AdminController@getGuests')->name('getGuests');` // executes `getGuests()` in `AdminController`.

29. `Route::get('/Admin/logout','AdminController@logout')->name('logout');` // executes `logout()` in `AdminController`.

### F.1 User Profile

30. `Route::get('/Admin/profile/{id}','AdminController@showUserProfile')->name('showUserProfile');` // executes `showUserProfile()` in `AdminController`.

31. `Route::get('/Admin/profile/{id}/edit','AdminController@editUserProfile')->name('editUserProfile');` // executes `editUserProfile()` in `AdminController`.

32. `Route::post('/Admin/profileUpdate','AdminController@updateEmployeeDetails')->name('updateEmployeeDetails');` // executes `updateEmployeeDetails()` in `AdminController`.

33. `Route::get('/Admin/profile/{id}/deactivate','AdminController@deactivate')->name('deactivate');` // executes `deactivate()` in `AdminController`.

34. `Route::get('/Admin/profile/{id}/activate','AdminController@activate')->name('activate');` // executes `activate()` in `AdminController`.

35. `Route::post('/Admin/profile/{id}/access-right','AdminController@giveAccessRights')->name('giveAccessRights');`

### F.2 Change Password

36. `Route::get('/Admin/passwordChangeView','AdminController@password_change_view')->name('admin_password_change_view');` // executes `password_change_view()` in `AdminController`.

37. `Route::post('/Admin/passwordChange','AdminController@Admin_password_change')->name('Admin_password_change');` // executes `Admin_password_change()` in `AdminController`.

### F.3 OrgGroups

38. `Route::get('/Admin/orgGroups','AdminController@OrgGroupsName')->name('OrgGroupsView');` // executes `OrgGroupsName()` in `AdminController`.

39. `Route::get('/Admin/orgGroups/{group_dn}','AdminController@getOrgGroupInfo')->name('getGroupInfoView');` // executes `getOrgGroupInfo()` in `AdminController`.

40. `Route::post('/Admin/orgGroups/addGroup','AdminController@addGroup')->name('addGroup');` // executes `addGroup()` in `AdminController`.

///40. `Route::post('/Admin/getUsers','AdminController@getUsersName')->name('getUsers');`

41. `Route::post('/Admin/orgGroups/rename','AdminController@renameGroupName')->name('renameGroupName');` // executes `renameGroupName()` in `AdminController`.

42. `Route::post('/Admin/orgGroups/changeOwnerName','AdminController@changeOwner')->name('changeOwner');` // executes `changeOwner()` in `AdminController`.

## F.7. Security Groups

43. `Route::get('/Admin/SecurityGroups','AdminController@SecurityGroupNames')->name('SecurityGroupNames');` // executes `SecurityGroupNames()` in `AdminController`.

44. `Route::get('/Admin/securityGroups/{group_dn}','AdminController@SecurityGroupInfo')->name('getHostInfoView');` // executes `SecurityGroupInfo()` in `AdminController`.

44. `Route::post('/Admin/securityGroups/addHost','AdminController@addHost')->name('addHost');` // executes `addHost()` in `AdminController`.

46. `Route::post('/Admin/securityGroups/rename','AdminController@renameHostName')->name('renameHostName');` // executes `renameHostName()` in `AdminController`.

## 6.8. Common method for OrgGroup and Security Groups

47. `Route::get('/Admin/addMember/{username}','AdminController@addMember')->name('addMember');` // executes `addMember()` in `AdminController`.

48. Route::post('/Admin/removeMember/{commanname}','AdminController@removeMember')->name('removeMember'); // executes removeMember() in AdminController.

#### F.9 Deadline

49. Route::get('/Admin/deadline','AdminController@deadline')->name('deadline');

#### F.10 Log Viewer

50. Route::get('logs','\Rap2hpoutre\LaravelLogViewer\LogViewerController@index')->name('logviewer'); // executes index() in LogViewerController.

### G. User Request Management

51. Route::get('/head/DivHeadApprovalView','EmployeeController@divHeadApprovalView')->name('divHeadApprovalView'); //executes divHeadApprovalView() in EmployeeController.

52. Route::post('/head/DivHeadApproval','EmployeeController@divHeadApproval')->name('divHeadApproval'); //executes divHeadApproval() in EmployeeController.

53. Route::post('/head/DivHeadDisApproval','EmployeeController@divHeadDisApproval')->name('divHeadDisApproval'); //executes divHeadDisApproval() in EmployeeController.

54. Route::get('/CCHead/CCUserApproval','EmployeeController@CCHeadApprovalView')->name('CCHeadApprovalView'); // executes CCHeadApprovalView() in EmployeeController.

55. Route::post('/CCHead/DivHeadApproval','EmployeeController@CCHeadApproval')->name('CCHeadApproval'); //executes CCHeadApproval() in EmployeeController.

56. Route::post('/CCHead/DivHeadDisApproval','EmployeeController@CCHeadDisApproval')->name('CCHeadDisApproval'); // executes CCHeadDisApproval() in EmployeeController.

● IndexController :- contains following functionalities :

1. checkUsername() :- This function return true if username already exists else nothing.
2. getDivisionHead() :- This function returns all Owners of group in json\_encode form.
3. getDepartmentOfOwner() :- This function returns all group name in json\_encode form associated with Owner.
4. getUID() :- if input is in form ' Mr. Pankaj Patel (pankajpatel)' then this function returns pankajpatel as output.
5. getPasswordChangeView() :- This function return view named passwordchange lying in views/index.
6. passwordchange() :- This function executes on submit of view named passwordchange lying in views/index, first checks user's old password, if incorrect then return back else further modifies user's password with new password.
7. checkOldpassword() :- This function checks user's password if correct then return true else false.
8. getPasswordResetEmail() :- This function returns view named ResetPasswordEmailView lying in views/index which takes username.
9. sendResetPasswordLink() :- This function executes on submit of view named first ResetPasswordEmailView lying in views/index, first fetches user's email from ldap using username. To generate random token, user's email is encoded in base64\_encode and stored into email\_token variable. UserId and email\_token is inserted into User table. Then view named ResetPasswordEmailLink lying in views/index is sent on to user's email.
10. passwordResetEmailView() :- This functions returns view named ResetPasswordLinkView lying in views/index when user clicks on link sent on to user's mail.
11. changePasswordEmail() :- This functions executes when user clicks on to change password button and updates user's password with new password.
12. getStatusView() :- This function returns view named statusView lying in resources/views/index/status.



13. getStatus() :- This function returns view named status lying inside index/status with required information.

- LoginController :- contains following functionalities :

14. Login() : login.blade.php lying in views/auth which takes username, password and captcha, On login this function is executed. This function checks for valid captcha, authenticates user input data with ldap, checks whether user is Admin or not if user is admin then 'admin\_username' is set into session else set 'username'. Then user with valid credentials redirects to '/home'. '/home' means index() in HomeController is executed.

15. Logout(): flush session contents and redirects to '/login'.

- HomeController (Non - Admin):- contains following functionalities :-

16. checkGroupMember() :- this function checks whether logged in user is Owner of any group, if yes then set 'ismember' equals true in session else do nothing. Then checks whether logged user is CC-Head, if yes then set 'cc\_head' equals true in session else do nothing.

17. index() :- This function first executes checkGroupMember(), checks if admin\_username exists in session object then redirect to '/Admin/home'. Here '/Admin/home' this route executes index() in AdminController. Else checks if username in session object then redirect to view('Employee.display') with user\_attributes variable.

18. getUserDetails() :- This function returns user details fetched from ldap server;

19. getUserInfo() :- This function executes getUserDetails() to retrieve user detail from ldap, then User Object is created by passing user detail and builder object as parameter to Model constructor. General, Personal, OrgGroup and Access-right information is added to array named user\_attributes. At last user\_attributes is return from this function.

20. upload() :- This function uploads user photo to ldap. User Photo is converted into string using file\_get\_contents function.

21. photo() :- This function retrieves user photo from ldap. From ldap, photo is retrieved in form of string, so converted into image using imagecreatefromjpeg function.

22. Search() :- This function returns all users having common name begins with input search value.
23. Logout() :- This function flush out session contents and redirect to login page.
24. Profile() :- This function display searched user profile.
25. password change view() :- This function returns password change view lying in resource/views/Employee.

● EmployeeController :- contains following functionalities:-

26. Index() :- This function returns view named create lying inside Employee/create.
27. store() :- This function executes on submit of view named create lying inside Employee, all attributes value come from view are validated and inserted into LDAP. If Username already exists returns error message. Once user's information inserted into LDAP then mail is sent to Division Head using sendMailToDivHead(). If everything have went correct return with message 'Your Registration is Successful' else returns message with 'Your Registration is Unsuccessful Contact Administrator'.
28. edit() :- This function takes userid as input, returns view named editProfile lying inside Employee along with User Object.
29. update() :- This function executes on submit of view named editProfile lying inside Employee, if modified information into ldap then returns user's homepage else display error message.
30. changePassword() :- This function executes on submit of view named password change view lying in resource/views/Employee, first checks user's old password using checkOldPassword(). If oldPassword is true then modify user's old password with new password.
31. SendMailToDivHead() :- This function first fetches cc-head's email from mysql table named user cc head approval and update user div head approval table with required information. Then send Mail to div-head for user's request approval.
32. divHeadApprovalView() :- This function returns view named userDivHeadApproval lying in index/DivHeadApproval along with approvals list information.

33. divHeadApproval() :- This function executes through ajax, updates div\_head\_status to 'Approved' in user\_div\_head\_approval table and send mail to chead using sendMailToCCHead(). If everything went correct returns true else false.
34. divHeadDisApproval() :- This function executes through ajax, updates div\_head\_status to 'Disapproved' in user\_div\_head\_approval table. If everything went correct returns true else false.
35. CCHeadApprovalView() :- This function fetches all data from user\_div\_head\_approval table and returns view named userCCHeadApproval lying in index/CCHeadApproval with fetched information.
36. CCHeadApproval() :- This function executes through Ajax, updates cc\_head\_approval\_status to 'Approved' in user\_div\_head\_approval table. Then mail is send to Computer Division using SendMailToComputerDivision() and send temporary password on to user's mail using sendPassword(). If everything went correct then returns true else false.
37. CCHeadDisApproval() :- This function executes through Ajax, updates cc\_head\_approval\_status to 'Disapproved' in user\_div\_head\_approval table. If everything went correct then returns true else false.
38. SendMailToComputerDivision() :- This function sends mail to computer division.
39. sendPassword() :- This function sends temporary password through mail on to user's mail.
40. createRandomString() :- This function generates random string and returns random string.

● AdminController :- Contains following functionalities :

41. index() :- This function returns view named home lying inside Admin with necessary information.
42. totalUsers():- This function returns total registered users.
43. lockedUsers():- This function returns total locked users.
44. unlockedUsers():- This function returns total unlocked users.
45. getUsersName() :- This function executes through Ajax, returns all users name in JSON format.

46. logout() :- This function flush out session content and redirect to login page.
47. getAllUsers():- This function returns view named allusers lying in views/Admin with necessary information.
48. deleteUsers() :- This function deletes selected user and redirect view to allusers lying in Admin
49. getLockedUsers():- This function returns view named lockedusers lying in views/Admin with necessary information.
50. getUnLockedUsers():- This function returns view named unlockedusers lying in views/Admin with necessary information.
51. getGuests() :- This function returns view name guests lying in views/guests with necessary information.
52. createUser():- This function creates user's account with limited fields.
53. showUserProfile() :- This function returns view named userprofile lying in views/Admin with all selected user related information.
54. giveAccessRights() :- This function executes through Ajax, returns 'add' if role is added\_successfully and returns 'remove' if role is removed successfully.
55. editUserProfile():- This function returns view named editprofile lying in views/Admin with necessary user information.
56. updateEmployeeDetails() :- This function executes on submit of view named editprofile lying in views/Admin and updates user information and redirects to user profile page.
57. activate() :- This function takes username as input parameter and activates user's account and redirects to user profile.
58. deactivate() :- This function takes username as input parameter and deactivates user's account and redirects to user profile.
59. password\_change\_view() :- This function returns view named passwordchangeview lying in view/Admin.
60. Admin\_password\_change() :- This function executes on submit of view named passwordchangeview lying in view/Admin, reset user's password with new password and redirects to same page.

61. OrgGroupsName() :- This function returns view named group lying in views/Admin with necessary group information (Displays all Groups with Owner).
62. getOrgGroupInfo() :- This function returns view named groupinfo lying in views/Admin with necessary information (Displays information of selected group and also its subgroup if any);
63. addGroup() :- This function adds new group and redirects to OrgGroupName page.
64. renameGroupName() :- This function rename existing group name and redirects to same group.
65. SecurityGroupNames():-This function returns view named hosts lying in views/Admin with necessary host information (Displays all hosts).
66. SecurityGroupInfo() :- This function returns view named hostinfo lying in views/Admin with necessary information (Displays information of selected host and also its subhosts if any);
67. addHost() :- This function adds new host and redirects to security groups page.
68. renameHostName() :- This function rename existing host name and redirects to same host page.
69. changeOwner():- This function changes owner of group and redirects to group info page.
70. addMember():- This function add selected member to Group or Host and redirects to their group or host info page.
71. removeMember():- This function remove selected member from Group or Host and redirects to their group or host info page.
72. deadline():- This function returns view named deadline lying in views/Admin with previous month and current month disabled users,

