public *function* phoneRegister(*Request* $request)

{

try {

$rules = [

"phone" => "required|unique:users,phone",

];

$validator = *Validator*::make($request->all(), $rules);

if($validator->fails()) {

$status = false;

$responseCode = 550;

$responseData = '';

$responseMsg = $validator->errors()->all()[0];

return response() -> json(["status"=>$status,"code"=>$responseCode,"data"=>$responseData,"message"=>$responseMsg]);

}

$digits = 4;

// $signup\_otp = rand(pow(10, $digits - 1), pow(10, $digits) - 1);

$signup\_otp = '1234';

if($request->refer\_code != ''){

$code = base64\_decode($request->refer\_code);

$tempuser = *TempUser*::create([

'phone' => $request->phone,

'otp' => $signup\_otp,

'refer\_id' => $code

]);

}

else{

$tempuser = *TempUser*::create([

'phone' => $request->phone,

'otp' => $signup\_otp

]);

}

if (!is\_null($tempuser)) {

$status = true;

$responseCode = 200;

$responseData = $tempuser;

$responseMsg = "OTP is send to your registered phone number";

} else {

$status = false;

$responseCode = 201;

$responseData = $user;

$responseMsg = "Failed to register";

}

} catch (\*Exception* $e) {

*Log*::error(" :: EXCEPTION :: " . $e->getMessage() . "\n" . $e->getTraceAsString());

$status = false;

$responseData = NULL;

$responseCode = $e->getCode();

$responseMsg = $e->getMessage();

}

return response() -> json(["status"=>$status,"code"=>$responseCode,"message"=>$responseMsg,"data"=>$responseData]);

}

/\*\*

\* Verify Signup Otp

\* @bodyParam user\_id number required Example: 2

\* @bodyParam otp number required Example: 1234

\* @response {

"status": true,

"code": 200,

"message": "Otp verified",

"data": {

"id": 2,

"phone": "8978675645",

"otp": "",

"password": null,

"first\_name": null,

"last\_name": null,

"gender": null,

"date\_of\_birth": null,

"about\_me": null,

"counrty\_code": null,

"created\_at": "2022-03-23T07:10:08.000000Z",

"updated\_at": "2022-03-23T07:32:34.000000Z"

}

}

\*/

public *function* verifySignupOtp(*Request* $request)

{

try {

$rules = [

"user\_id" => "required",

"otp" => "required|min:4"

];

$validator = *Validator*::make($request->all(), $rules);

if($validator->fails()) {

$status = false;

$responseCode = 550;

$responseData = '';

$responseMsg = $validator->errors()->all()[0];

return response() -> json(["status"=>$status,"code"=>$responseCode,"data"=>$responseData,"message"=>$responseMsg]);

}

$tempuser = *TempUser*::find($request->user\_id);

if ($request->otp == $tempuser->otp) {

$tempuser->otp = '';

$tempuser->save();

$status = true;

$responseCode = 200;

$responseData = $tempuser;

$responseMsg = "Otp verified";

} else {

$status = false;

$responseCode = 201;

$responseData = $tempuser;

$responseMsg = "Otp does not match";

}

} catch (\*Exception* $e) {

*Log*::error(" :: EXCEPTION :: " . $e->getMessage() . "\n" . $e->getTraceAsString());

$status = false;

$responseData = NULL;

$responseCode = $e->getCode();

$responseMsg = $e->getMessage();

}

return response() -> json(["status"=>$status,"code"=>$responseCode,"message"=>$responseMsg,"data"=>$responseData]);

}

/\*\*

\* Create Signup Password

\* @bodyParam user\_id number required Example: 53

\* @bodyParam password string required Example: 11111111

\* @bodyParam confirm\_password string required Example: 11111111

\* @response {

"status": true,

"code": 200,

"message": "Password created successfully",

"data": {

"id": 2,

"phone": "8978675645",

"otp": "",

"password": "12345678",

"first\_name": null,

"last\_name": null,

"gender": null,

"date\_of\_birth": null,

"about\_me": null,

"counrty\_code": null,

"created\_at": "2022-03-23T07:10:08.000000Z",

"updated\_at": "2022-03-23T07:36:57.000000Z"

}

}

\*/

public *function* signupPassword(*Request* $request)

{

try {

$rules = [

"user\_id" => "required",

"password" => "required|min:6",

"confirm\_password" => "required|min:6",

];

$validator = *Validator*::make($request->all(), $rules);

if($validator->fails()) {

$status = false;

$responseCode = 550;

$responseData = '';

$responseMsg = $validator->errors()->all()[0];

return response() -> json(["status"=>$status,"code"=>$responseCode,"data"=>$responseData,"message"=>$responseMsg]);

}

$tempuser = *TempUser*::find($request->user\_id);

if (!is\_null($tempuser)) {

if($request->password == $request->confirm\_password){

$tempuser->fill($request->all());

$tempuser->save();

$status = true;

$responseCode = 200;

$responseData = $tempuser;

$responseMsg = "Password created successfully";

}

else{

$status = false;

$responseCode = 201;

$responseData = $tempuser;

$responseMsg = "Password and Confirm password does not match";

}

} else {

$status = false;

$responseCode = 201;

$responseData = $tempuser;

$responseMsg = "Failed to add password";

}

} catch (\*Exception* $e) {

*Log*::error(" :: EXCEPTION :: " . $e->getMessage() . "\n" . $e->getTraceAsString());

$status = false;

$responseData = NULL;

$responseCode = $e->getCode();

$responseMsg = $e->getMessage();

}

return response() -> json(["status"=>$status,"code"=>$responseCode,"message"=>$responseMsg,"data"=>$responseData]);

}

/\*\*

\* Agree with Rules

\* @bodyParam user\_id number required Example: 53

\* @response {

"status": true,

"code": 200,

"message": "Agreed with house rules",

"data": {

"id": 2,

"phone": "8978675645",

"otp": "",

"password": "12345678",

"first\_name": null,

"last\_name": null,

"gender": null,

"date\_of\_birth": null,

"about\_me": null,

"counrty\_code": null,

"created\_at": "2022-03-23T07:10:08.000000Z",

"updated\_at": "2022-03-23T07:36:57.000000Z"

}

}

\*/

public *function* agreeTerms(*Request* $request){

try {

$rules = [

"user\_id" => "required"

];

$validator = *Validator*::make($request->all(), $rules);

if($validator->fails()) {

$status = false;

$responseCode = 550;

$responseData = '';

$responseMsg = $validator->errors()->all()[0];

return response() -> json(["status"=>$status,"code"=>$responseCode,"data"=>$responseData,"message"=>$responseMsg]);

}

$tempuser = *TempUser*::find($request->user\_id);

if (!is\_null($tempuser)) {

$tempuser->is\_agreed = '1';

$tempuser->save();

$status = true;

$responseCode = 200;

$responseData = $tempuser;

$responseMsg = "Agreed with house rules";

}

else{

$status = false;

$responseCode = 201;

$responseData = $tempuser;

$responseMsg = "Donot agree with house rules";

}

}

catch (\*Exception* $e) {

*Log*::error(" :: EXCEPTION :: " . $e->getMessage() . "\n" . $e->getTraceAsString());

$status = false;

$responseData = NULL;

$responseCode = $e->getCode();

$responseMsg = $e->getMessage();

}

return response() -> json(["status"=>$status,"code"=>$responseCode,"message"=>$responseMsg,"data"=>$responseData]);

}

/\*\*

\* Final Registration and Login

\* @bodyParam user\_id number required Example: 53

\* @bodyParam first\_name string required Example: Jhon

\* @bodyParam last\_name string optional Example: Doe

\* @bodyParam gender string required male,female Example: male

\* @bodyParam date\_of\_birth optional Example: 1999-02-09

\* @bodyParam about\_me optional Example: good human

\* @bodyParam sextual\_orientation optional Example: straight

\* @bodyParam show\_orientation optional Example: 1

\* @response {

"status": true,

"code": 200,

"message": "Final Registration done & Login successfully",

"data": {

"id": 54,

"first\_name": "duke",

"last\_name": null,

"email": null,

"phone": "8978675645",

"address": null,

"email\_verified\_at": null,

"current\_team\_id": null,

"profile\_photo\_path": null,

"created\_at": "2022-03-23T08:29:55.000000Z",

"updated\_at": "2022-03-23T08:29:55.000000Z",

"date\_of\_birth": null,

"about\_me": null,

"gender": null,

"active": 1,

"signup\_otp": null,

"token": "1|whrxoqWoomxXRTVNtxbvR6DRMZSxfBKBYi8dxQTS",

"full\_name": "duke ",

"role\_name": 0,

"profile\_photo\_url": "https://ui-avatars.com/api/?name=duke&color=7F9CF5&background=EBF4FF",

"profile\_image\_url": null,

"media": []

}

}

\*/

public *function* finalRegister(*Request* $request)

{

try {

$rules = [

"user\_id" => "required",

"first\_name" => "required",

"gender" => "required"

];

$validator = *Validator*::make($request->all(), $rules);

if($validator->fails()) {

$status = false;

$responseCode = 550;

$responseData = '';

$responseMsg = $validator->errors()->all()[0];

return response() -> json(["status"=>$status,"code"=>$responseCode,"data"=>$responseData,"message"=>$responseMsg]);

}

$tempuser = *TempUser*::find($request->user\_id);

$phoneCount = *User*::where('phone',$tempuser->phone)->count();

if($phoneCount){

$status = false;

$responseCode = 201;

$responseData = [];

$responseMsg = "Phone number is already exist!!";

}

else{

$temp\_password = $tempuser->password;

if (!is\_null($tempuser)) {

$user = new *User*;

$user->first\_name = $request->first\_name;

$user->phone = $tempuser->phone;

$user->last\_name = $request->last\_name;

$user->gender = $request->gender;

$user->date\_of\_birth = $request->date\_of\_birth;

$user->about\_me = $request->about\_me;

$user->password = $tempuser->password;

$user->is\_agreed = $tempuser->is\_agreed;

$user->show\_orientation = $request->show\_orientation;

$user->sextual\_orientation = $request->sextual\_orientation;

$user->active = true;

if($tempuser->refer\_id != ''){

$user->refer\_id = $tempuser->refer\_id;

}

$user->save();

$user->assignRole('CLIENT');

//firebase authentication

$auth = *Firebase*::auth();

$token = $auth->createCustomToken('uid');

//return response()->json(["status"=>$token]);

$userProperties = [

'phoneNumber' => $tempuser->phone,

'password' => $tempuser->password,

'displayName' => $request->first\_name,

'disabled' => false,

];

$createdUser = $auth->createUser($userProperties);

$user->firebase\_token = $createdUser->uid;

$user->save();

$tempuser->delete();

if (*Auth*::attempt(['phone' => $user->phone, 'password' => $temp\_password])) {

$user = *Auth*::user();

$token = $user->createToken('token')->plainTextToken;

$user->token = $token;

$customToken = $auth->createCustomToken($user->firebase\_token);

$signInResult = $auth->signInWithCustomToken($customToken);

$status = true;

$responseCode = 200;

$responseData = $user;

$responseMsg = "Final Registration done & Login successfully";

} else {

$status = false;

$responseCode = 201;

$responseData = [];

$responseMsg = "Whoops! invalid password";

}

} else {

$status = false;

$responseCode = 201;

$responseData = $user;

$responseMsg = "Failed to register";

}

}

} catch (\*Exception* $e) {

*Log*::error(" :: EXCEPTION :: " . $e->getMessage() . "\n" . $e->getTraceAsString());

$status = false;

$responseData = NULL;

$responseCode = $e->getCode();

$responseMsg = $e->getMessage();

}

return response() -> json(["status"=>$status,"code"=>$responseCode,"message"=>$responseMsg,"data"=>$responseData]);

}