**Safecity Start**

**Onboarding**

<?php $this->load->view("includes/header"); ?>

<main class="contentCenter">

<div class="main-content mt-5">

<div class="container-fluid">

<div class="container">

<div class="text mx-auto shareIncidentHome onbarding">

<form>

<h1><?php **echo** $this->lang->line("select\_location"); ?></h1>

<div class="fix-height">

<!-- location start -->

<div class="form-group">

<label><?php **echo** $this->lang->line("select\_country"); ?></label>

<input type="text" class="form-control js-states" id="autocomplete-country" value="<?= $country\_name ?>" autocomplete="country\_id">

<input type="hidden" id="country\_id" name="country\_id" value="<?= $country\_id ?>">

<input type="hidden" id="ngo\_id" name="ngo\_id" value="<?= @$\_COOKIE["ngo\_id"]? $\_COOKIE["ngo\_id"]: 0 ?>">

</div>

<!-- location end -->

<!-- city start -->

<div class="form-group" id="cityblock">

<label for="exampleInputPassword1">

<?php **echo** $this->lang->line("select\_city"); ?>

</label>

<input type="text" class="form-control js-states" id="autocomplete-city" value="<?= $city\_name ?>" autocomplete="city\_id">

<input type="hidden" id="city\_id" name="city\_id" value="<?= $city\_id ?>">

</div>

<!-- city end -->

<!-- organization radio btn start -->

<div class="form-group" id="organizationradioblock">

<label><?php **echo** $this->lang->line("reporting\_orgnization"); ?></label>

<div class="clearfix"></div>

<div class="custom-control custom-radio custom-control-inline">

<input type="radio" class="custom-control-input" id="customRadio" name="is\_organization" value="1">

<label class="custom-control-label" for="customRadio">

<?php **echo** $this->lang->line("org\_yes"); ?>

</label>

</div>

<div class="custom-control custom-radio custom-control-inline">

<input type="radio" class="custom-control-input" id="customRadio2" name="is\_organization" value="0">

<label class="custom-control-label" for="customRadio2">

<?php **echo** $this->lang->line("org\_no"); ?>

</label>

</div>

</div>

<!-- organizationradio btn end -->

<!-- organization start -->

<div class="form-group" id="oraganisationblock">

<label><?php **echo** $this->lang->line("select\_organization"); ?></label>

<input type="text" class="form-control js-state" id="autocomplete-organization">

<input type="hidden" id="organization\_id">

</div>

<!-- organization end -->

<!-- language start -->

<!-- <div class="form-group" id="languageblock">

<label><?php **echo** $this->lang->line("select\_language"); ?></label>

<input type="text" class="form-control js-state" id="autocomplete-language" autocomplete="off">

<input type="hidden" id="language\_id">

</div> -->

<div class="form-group" id="languageblock">

<label><?php **echo** $this->lang->line("select\_language"); ?></label>

<select id="autocomplete-language-id" class="form-control js-state">

<option value="">Select Language</option>

</select>

<input type="hidden" id="language\_id">

</div>

<!-- language end -->

</div>

<!-- Proceed btn start -->

<div class="text-left">

<button disabled id="proceed\_btn" class="btn btn-primary safetyBtn animated nextPage fadeInUp pull-right mt-3 nxt\_btn" style="margin-left: 0!important">

<?php **echo** $this->lang->line("button\_proceed"); ?>

</button>

</div>

<!-- Proceed btn end -->

<div class="mt-5"></div>

</form>

</div>

</div>

</div>

</div>

</div>

</main>

<div class="modal fade org\_varification" id="org\_varification" tabindex="-1" role="dialog">

<div class="modal-dialog modal-dialog-centered mt-0" role="document">

<div class="modal-content">

<div class="modal-body">

<div class="form-group">

<label><?php **echo** $this->lang->line("organization\_code"); ?></label>

<input type="text" name="varification\_code" id="varification\_code" class="form-control">

<div class="text-danger"></div>

</div>

<div class="text-center">

<button type="button" id="org\_done" class="btn w-85 btn\_purple mb-0 mt-3 w-100">

<?php **echo** $this->lang->line("button\_done"); ?>

</button>

</div>

</div>

</div><!-- /.modal-content -->

</div><!-- /.modal-dialog -->

</div><!-- /.modal -->

<?php $this->load->view("includes/footer"); ?>

<style type="text/css">

.typeahead>li.active {

text-decoration: none;

background-color: #EBE2F4;

}

</style>

<script>

var baseUrl = '<?= base\_url() ?>';

var country\_id = <?= $country\_id ?>;

var city\_id = <?= $city\_id ?>;

var client\_id = <?= $client\_id ?>;

var organization\_id = language\_id = 0;

// language changes 29-10-2020 start - sonam

var code\_error = '<?php **echo** $this->lang->line("correct\_code"); ?>';

// language changes 29-10-2020 end - sonam

</script>

<script src="application/modules/report\_incident/scripts/onboarding.js"></script>

**Consent:**

<?php

$this->load->view('includes/header');

?>

<main>

<div class="main-content covering my-auto">

<div class="container">

<?php

*//echo $this->lang->line('menu\_lang');*

**if**(isset($consent\_data[0]['content']) && !**empty**($consent\_data[0]['content'])){

**echo** $consent\_data[0]['content'];

?>

<div class="text mx-auto shareIncident">

<div class="custom-control custom-checkbox agreeConsent" style="padding-left:24px!important"><input type="checkbox" name="agreeConsent" class="custom-control-input" id="agreeConsent"><label class="custom-control-label" for="agreeConsent"><?php **echo** $this->lang->line('apply\_to\_agreeConsent'); ?></label></div><div class="mt-4"><a href="privacy\_policy" class="themeColor aLink"><?php **echo** $this->lang->line('apply\_to\_privacy\_policy'); ?></a></div>

<div><button class="btn btn-primary safetyBtn consent\_btn animated nextPage fadeInUp pull-right mt-3 nxt\_btn" disabled="disabled" style="margin-left:0!important"><?php **echo** $this->lang->line('apply\_to\_next\_btn'); ?></button></div>

</div>

<?php

} **else** {

**echo** "Data not found.";

}

?>

</div>

</div>

</main>

<?php

$this->load->view('includes/footer');

?>

<script>

$(document).ready(function() {

if($("#agreeConsent").is(':checked')) {

$(".nxt\_btn").removeAttr('disabled');

} else {

$(".nxt\_btn").attr('disabled', 'disabled');

}

});

</script>

**Report an Incident**

<?php

defined('BASEPATH') **or** **exit**('No direct script access allowed');

$this->load->view('../includes/header');

?>

<body class="bgColor relativepo">

<div class="count float-right">

<h5 class="progress-text">0% Completed </h5>

<div class="progress">

<div class="progress-bar" style="width: 0%;"> </div>

</div>

</div>

<div class="covering my-auto">

<div class="container">

<div class="text mx-auto width95" style="width: 62%;">

<div class="questionaire">

<form>

<ul class="progress-form">

<div class="Step1">

<a id="dynamicBack" href="javascript:void(0);" class="pull-left animated fadeInUp dynamic\_back"><img src="<?php **echo** base\_url() ?>assets/images/icon-feather-arrow-left.svg" class="img-fluid leftIcon"></a>

<!-- Dynamic questions start -->

<li class="form-group animated fadeInUp activate selected singleColumn" data-color="#7C6992" data-percentage="100%">

<label>

<span id="question\_span"><!-- Placeholder for dynamic questions --></span><span class="error">\*</span><br>

<span class="themeColor" id="subtext\_span"><!-- Placeholder for dynamic sub text --></span>

</label>

<div id="options">

<!-- Placeholder for dynamic options -->

</div>

<span class="sharingForInfo"></span>

</li>

<!-- Dynamic questions end -->

</div>

</ul>

</form>

<button id="dynamicNext" class="btn btn-primary nxt\_btn animated nextPage fadeInUp pull-right mt-4" ><?php **echo** $this->lang->line('apply\_to\_next\_btn'); ?><span class="glyphicon glyphicon-chevron-down"></span></button>

</div> <!-- //questionaire -->

<div class="thankyou-section" style="display:none">

<div class="text mx-auto width100" style="width: 58%;">

<h4 class="mb-4 text-center textColor dynamic-success-title">Thank you for submitting your report!</h4>

<div class="text-center">

<img src="<?php **echo** base\_url() ?>assets/images/thank-you-check.png" class="img-fluid checkIcon">

</div>

<div class="text mb-5 dynamic-success-content">

<p>By sharing your experience with us, you are helping prevent 3 others from experiencing something similar.</p>

<p>If you have 5-10 minutes, we would like to know more about the incident to understand other factors that play a role in sexual violence. By answering a few questions, you will help us build safer cities.</p>

</div>

<div class="text-center mt-4 dynamic-success-link">

<a href="#" class="btn w-75 width100 btn\_p\_white mb-4">YES, I WANT TO HELP</a>

<a href="home" class="btn w-75 width100 btn\_purple ">NO, I'M DONE</a>

</div>

</div>

</div>

</div> <!-- //text mx-auto -->

</div>

<div class="clearfix"></div>

</div>

<?php $this->load->view('../includes/footer'); ?>

<script src="https://maps.google.com/maps/api/js?key=AIzaSyA-RG4hM7qRh3jHfOwSuUOBexPTn0CZf6w&libraries=geometry,places" async></script>

<script type="text/javascript">

var baseURL = "<?php **echo** base\_url() ?>";

var clientForms = <?php **echo** $forms ?>;

var dynamicQuestionOptionJson = <?php **echo** $questions ?>;

var categories = <?php **echo** $categories ?>;

var lang\_id = $.cookie("language\_id") || 1;

//Language based text

var dateform\_select\_date = "<?php **echo** $this->lang->line('dateform\_select\_date'); ?>";

var this\_is\_an\_estimate = "<?php **echo** $this->lang->line('this\_is\_an\_estimate'); ?>";

var timeform\_select\_time = "<?php **echo** $this->lang->line('timeform\_select\_time'); ?>";

var timeform\_select\_a\_time\_range = "<?php **echo** $this->lang->line('timeform\_select\_a\_time\_range'); ?>";

var timeform\_clear\_time = "<?php **echo** $this->lang->line('timeform\_clear\_time'); ?>";

var timeform\_clear\_time\_range = "<?php **echo** $this->lang->line('timeform\_clear\_time\_range'); ?>";

var timeform\_time\_or = "<?php **echo** $this->lang->line('timeform\_time\_or'); ?>";

var address\_locate\_on\_map = "<?php **echo** $this->lang->line('address\_locate\_on\_map'); ?>";

var address\_locate\_on\_map\_subtext = "<?php **echo** $this->lang->line('address\_locate\_on\_map\_subtext'); ?>";

var address\_pin\_exact\_location = "<?php **echo** $this->lang->line('address\_pin\_exact\_location'); ?>";

var map\_start\_typing = "<?php **echo** $this->lang->line('map\_start\_typing'); ?>";

var address\_pin\_exact\_location\_subtext = "<?php **echo** $this->lang->line('address\_pin\_exact\_location\_subtext'); ?>";

var address\_pinned\_on\_map = "<?php **echo** $this->lang->line('address\_pinned\_on\_map'); ?>";

var address\_area = "<?php **echo** $this->lang->line('address\_area'); ?>";

var address\_area\_subtext = "<?php **echo** $this->lang->line('address\_area\_subtext'); ?>";

var address\_build\_loc = "<?php **echo** $this->lang->line('address\_build\_loc'); ?>";

var address\_build\_loc\_subtext = "<?php **echo** $this->lang->line('address\_build\_loc\_subtext'); ?>";

var address\_i\_confirm\_msg = "<?php **echo** $this->lang->line('address\_i\_confirm\_msg'); ?>";

var address\_i\_confirm = "<?php **echo** $this->lang->line('address\_i\_confirm'); ?>";

var someone\_else\_popup\_msg = "<?php **echo** $this->lang->line('someone\_else\_popup\_msg'); ?>";

var someone\_else\_popup\_ok = "<?php **echo** $this->lang->line('someone\_else\_popup\_ok'); ?>";

var required\_field = "<?php **echo** $this->lang->line('required\_field'); ?>";

</script>

<script src="<?php **echo** base\_url() ?>application/modules/report\_incident/scripts/dynamic\_forms.js"></script>

<!-- Load Custom Components -->

<script src="<?php **echo** base\_url() ?>application/modules/report\_incident/scripts/components/TextField.js"></script>

<script src="<?php **echo** base\_url() ?>application/modules/report\_incident/scripts/components/RadioField.js"></script>

<script src="<?php **echo** base\_url() ?>application/modules/report\_incident/scripts/components/CheckboxField.js"></script>

<script src="<?php **echo** base\_url() ?>application/modules/report\_incident/scripts/components/EstimateTimeRangePickerField.js"></script>

<script src="<?php **echo** base\_url() ?>application/modules/report\_incident/scripts/components/EstimateDatepickerField.js"></script>

<script src="<?php **echo** base\_url() ?>application/modules/report\_incident/scripts/components/AddressFormField.js"></script>

<script src="<?php **echo** base\_url() ?>application/modules/report\_incident/scripts/components/LocationPinMapField.js"></script>

</body>

</html>

**Onboarding JS**

$(**function** () {

**var** language\_text = 'English';

**var** language\_sf = "EN";

**var** language\_val = "English";

*// Hide all blocks at start*

**if** (country\_id == 0) $("#cityblock").hide();

*// Skipping organization selection*

*// until we onboard a client*

*/\*if (city\_id == 0) $("#organizationradioblock").hide();*

*$("#oraganisationblock").hide();*

*$("#languageblock").hide();\*/*

*// Remove this when we get a client*

**if** (city\_id == 0) {

$("#languageblock").hide();

} **else** {

setLanguages();

}

$("input[name=is\_organization]#customRadio2").prop('checked', **true**)

$("#organizationradioblock").hide();

$("#oraganisationblock").hide();

*// On organization change*

$("input[name=is\_organization]").click(**function** (event) {

**var** is\_organization = $("input[name=is\_organization]:checked").val();

**if** (

is\_organization == "1" &&

$("#oraganisationblock").css("display") == "none"

) {

hideLanguageBlock();

disableProceedButton();

$("#oraganisationblock").show();

} **else** **if** (

is\_organization == "0" &&

($("#oraganisationblock").css("display") != "none" ||

$("#languageblock").css("display") == "none")

) {

client\_id = 1;

hideOrganizationBlock();

hideLanguageBlock();

disableProceedButton();

$("#languageblock").show();

setLanguages();

}

});

**function** setLanguages() {

$.ajax({

url: baseUrl + "api/get-languages-autocomplete",

type: 'POST',

dataType: 'json',

data: {client\_id: client\_id},

})

.done(**function**(data) {

**var** html = '<option value="">Select Language</option>';

**if** (data.status == **true**) {

$("#autocomplete-language-id").empty();

data.data.forEach(**function**(language) {

html += '<option data-shortname="'+language.short\_name+'" data-nativename="'+language.native\_name+'" value="'+language.id+'">'+language.name+'</option>';

});

}

$("#autocomplete-language-id").html(html);

$("#autocomplete-language-id").select2({

placeholder: "Select a language"

});

})

.fail(**function**() {

console.log("error");

$("#autocomplete-language-id").html('');

$("#autocomplete-language-id").select2({

placeholder: "Select a language"

});

})

.always(**function**() {

console.log("complete");

});

}

$(document).on('change', '#autocomplete-language-id', **function**(event) {

event.preventDefault();

language\_id = $(**this**).val();

$("#language\_id").val(language\_id);

**if**(language\_id!='') {

**var** selectedData = $(**this**).select2('data');

**var** $selectedElem = $("#autocomplete-language-id option:selected");

**if**(selectedData!=**null**) {

language\_text = selectedData[0].text;

language\_sf = $selectedElem.data('shortname');

language\_val = $selectedElem.data('nativename');

} **else** {

language\_text = 'English';

language\_sf = 'EN';

language\_val = 'English';

}

enableProceedButton();

}

**else**

disableProceedButton();

});

$("#org\_done").click(**function** (event) {

event.preventDefault();

*// Reset passcode error message*

$("#varification\_code").siblings(".text-danger").html("");

**var** verification\_code = $("#varification\_code").val();

*// Check for code validity*

$.ajax({

url: baseUrl + "api/organization/verify-passcode",

type: "POST",

data: { id: organization\_id, passcode: verification\_code },

})

.done(**function** (response) {

console.log("success");

**if** (response.is\_valid) {

$("#org\_varification").modal("hide");

$("#languageblock").show();

setLanguages();

} **else** {

*// Set passcode error message*

$("#varification\_code")

.siblings(".text-danger")

.html(code\_error); *//code changes by sonam - 29-10-2020*

}

})

.fail(**function** () {

console.log("error");

})

.always(**function** () {

console.log("complete");

});

});

*/\*\* On submit/complete \*/*

$("#proceed\_btn").click(**function** (event) {

event.preventDefault();

**if** (client\_id > 0 && language\_id > 0) {

**var** country = $("#autocomplete-country").val();

**var** city = $("#autocomplete-city").val();

$.cookie("country", country);

$.cookie("country\_id", country\_id);

$.cookie("city", city);

$.cookie("city\_id", city\_id);

$.cookie("client\_id", client\_id);

$.cookie("language\_id", language\_id);

$.cookie("language", language\_text);

$.cookie("languageSF", language\_sf);

$.cookie("language\_val", language\_val);

**var** ngo\_id= $("#ngo\_id").val();

$.cookie("ngo\_id", ngo\_id);

*// Reset Form*

localStorage.removeItem("clientForms");

localStorage.removeItem("dynamicQuestionOptionJson");

localStorage.removeItem("incident\_id");

localStorage.removeItem("currentQuestion");

localStorage.removeItem("currentParentKey");

localStorage.removeItem("currentTreeOptions");

localStorage.removeItem("selectedAnswers");

localStorage.removeItem("dynamicQuestionJson");

localStorage.removeItem("currentForm");

localStorage.removeItem("dependedQuestionsAnswers");

localStorage.removeItem("totalQuestions");

localStorage.removeItem("totalParentAnswers");

localStorage.removeItem("isThankYouPage");

*// Get latitude/longitude*

**var** geocoder = **new** google.maps.Geocoder();

geocoder.geocode( { 'address': city+', '+country}, **function**(results, status) {

**if** (status == google.maps.GeocoderStatus.OK) {

$.cookie('lat', results[0].geometry.location.lat());

$.cookie('lng', results[0].geometry.location.lng());

**if**(ngo\_id > 0){

window.location.href = baseUrl + "ngo";

}**else**{

window.location.href = baseUrl + "shareIncident";

}

} **else** {

console.log('failed to get lat lng with status', status);

**if**(ngo\_id > 0){

window.location.href = baseUrl + "ngo";

}**else**{

window.location.href = baseUrl + "shareIncident";

}

}

});

} **else** {

alert("Please select all fields properly to proceed!");

}

});

*/\*\* Autocompletes \*/*

$("#autocomplete-country").typeahead({

autoSelect: **true**,

highlight: **true**,

hint: **true**,

displayText: **function** (item) {

**return** item.country\_name;

},

afterSelect: **function** (item) {

**this**.$element[0].value = item.country\_name;

country\_id = item.country\_id;

$("#country\_id").val(item.country\_id);

*// Reset Corresponding UI*

hideCityBlock();

$("#cityblock").show();

hideIsOrganizationBlock();

hideOrganizationBlock();

hideLanguageBlock();

disableProceedButton();

},

source: **function** (query, process) {

$.ajax({

url: baseUrl + "home/getCountryAutocomplete",

data: { query: query },

dataType: "json",

type: "POST",

success: **function** (data) {

process(data);

},

});

},

});

$("#autocomplete-country").change(**function**(event) {

console.log('in country changed');

**var** current = $(**this**).typeahead("getActive");

console.log(current);

**if**(!current || current.country\_name!=$(**this**).val()) {

*// Nothing is active so it is a new value (or maybe empty value)*

hideCityBlock();

hideIsOrganizationBlock();

hideOrganizationBlock();

hideLanguageBlock();

disableProceedButton();

}

});

$("#autocomplete-city").typeahead({

autoSelect: **true**,

highlight: **true**,

hint: **true**,

displayText: **function** (item) {

**return** item.city\_name;

},

afterSelect: **function** (item) {

console.log('fired!');

**this**.$element[0].value = item.city\_name;

$("#city\_id").val(item.id);

city\_id = item.id;

client\_id = item.client\_id;

*//added ngo*

ngo\_id=item.ngo\_id;

$("#ngo\_id").val(item.ngo\_id);

*// Reset Corresponding UI*

hideIsOrganizationBlock();

hideOrganizationBlock();

hideLanguageBlock();

disableProceedButton();

*// Skipping organization selection*

*// until we onboard a client*

*//$("#organizationradioblock").show();*

$("#languageblock").show();

setLanguages();

},

source: **function** (query, process) {

$.ajax({

url: baseUrl + "api/get-cities-autocomplete",

data: { query: query, country\_id: country\_id },

dataType: "json",

type: "POST",

success: **function** (data) {

**if** (data.status == **true**) process(data.data);

**else** process([]);

},

});

},

});

$("#autocomplete-city").change(**function**(event) {

**var** current = $(**this**).typeahead("getActive");

**if**(!current || current.city\_name!=$(**this**).val()) {

*// Nothing is active so it is a new value (or maybe empty value)*

hideIsOrganizationBlock();

hideOrganizationBlock();

hideLanguageBlock();

disableProceedButton();

}

});

$("#autocomplete-organization").typeahead({

autoSelect: **true**,

highlight: **true**,

hint: **true**,

displayText: **function** (item) {

**return** item.name;

},

afterSelect: **function** (item) {

**this**.$element[0].value = item.name;

$("#organization\_id").val(item.id);

organization\_id = item.id;

client\_id = item.client\_id;

*// Reset Corresponding UI*

hideLanguageBlock();

disableProceedButton();

**if** (item.has\_passcode == "1") {

$("#org\_varification").modal("show");

}

**else** {

$("#languageblock").show();

setLanguages();

}

},

source: **function** (query, process) {

$.ajax({

url: baseUrl + "api/get-organizations-autocomplete",

data: {

query: query,

country\_id: country\_id,

city\_id: city\_id,

},

dataType: "json",

type: "POST",

success: **function** (data) {

**if** (data.status == **true**) process(data.data);

**else** process([]);

},

});

},

});

$("#autocomplete-language").typeahead({

autoSelect: **true**,

highlight: **true**,

hint: **true**,

displayText: **function** (item) {

**return** item.name;

},

afterSelect: **function** (item) {

**this**.$element[0].value = item.name;

$("#language\_id").val(item.id);

language\_id = item.id;

enableProceedButton();

},

source: **function** (query, process) {

$.ajax({

url: baseUrl + "api/get-languages-autocomplete",

data: { query: query, client\_id: client\_id },

dataType: "json",

type: "POST",

success: **function** (data) {

**if** (data.status == **true**) process(data.data);

**else** process([]);

},

});

},

});

*/\*\* Set Visibilities \*/*

**function** hideCityBlock() {

city\_id = 0;

client\_id = 1;

$("#cityblock").hide();

$("#autocomplete-city").val("");

$("#city\_id").val("");

}

**function** hideIsOrganizationBlock() {

$("#organizationradioblock").hide();

$("input[name=is\_organization]").prop("checked", **false**);

}

**function** hideOrganizationBlock() {

organization\_id = 0;

$("#oraganisationblock").hide();

$("#autocomplete-organization").val("");

$("#organization\_id").val("");

}

**function** hideLanguageBlock() {

language\_id = 0;

$("#languageblock").hide();

$("#autocomplete-language").val("");

$("#language\_id").val("");

}

**function** enableProceedButton() {

$("#proceed\_btn").addClass("purple-btn");

$("#proceed\_btn").removeAttr("disabled");

}

**function** disableProceedButton() {

$("#proceed\_btn").removeClass("purple-btn");

$("#proceed\_btn").attr("disabled", "disabled");

}

});

**Dynamic Form**

**var** incident\_id = 0;

**var** currentQuestion = "";

**var** currentParentKey = 0;

**var** currentTreeOptions = [];

**var** selectedAnswers = [];

**var** dynamicQuestionJson = [];

**var** currentForm = 0;

**var** dependedQuestionsAnswers = {};

**var** isThankYouPage = **false**;

**var** totalQuestions = 0;

**var** totalParentAnswers = 0;

$(document).ready(**function** () {

$("#dynamicNext").attr("disabled", "disabled");

*// On page refresh start from where user left*

fetchState();

$("#dynamicBack").click(**function** (event) {

event.preventDefault();

$("#dynamicNext").removeAttr("disabled");

**if** (selectedAnswers.length == 0) {

**if**(top.location.pathname.includes('shareIncident-form')) {

*// Redirect back to actual previous page*

window.location.href = baseURL + "shareIncident";

} **else** {

*// Page is loaded in iframe for admin*

top.location.href = baseURL+"admin/incidents";

}

} **else** {

*// Back to previous question*

*// Get previous question back*

**var** lastQuestion = selectedAnswers.pop();

**if**(clientForms[currentForm].type=='primary' || currentParentKey!=lastQuestion.currentQuestion.currentParentKey) {

console.log('reducing progress -1');

*// Update answer*

totalParentAnswers--;

setProgress();

}

currentParentKey =

lastQuestion.currentQuestion.currentParentKey;

currentTreeOptions =

lastQuestion.currentQuestion.currentTreeOptions;

*// Has form changed to previous form?*

**if**(currentForm!=lastQuestion.currentQuestion.currentForm) {

console.log('form changed!');

currentForm = lastQuestion.currentQuestion.currentForm;

setFormDynamicQuestion(clientForms[currentForm], **function**() {

*// Show the previous question*

getDynamicQuestions(

lastQuestion.currentQuestion.id,

lastQuestion

);

});

} **else** {

*// Show the previous question*

getDynamicQuestions(

lastQuestion.currentQuestion.id,

lastQuestion

);

}

}

});

});

*/\*\* Progress Bar \*/*

**function** calculateTotalQuestions() {

totalQuestions = 0;

**for** (**var** i = 0; i < clientForms.length; i++) {

**if**(clientForms[i].type=='primary')

totalQuestions += JSON.parse(clientForms[i].question\_ids\_json).length;

**else** **if**(clientForms[i].type=='logic') {

console.log('calculate Total Questions:');

console.log(clientForms[i].estimate\_question);

totalQuestions += clientForms[i].estimate\_question || 0;

}

**if**(clientForms[i].is\_submit==**true**)

**break**;

}

console.log(totalQuestions);

setProgress();

}

**function** setProgress() {

**var** progressPercent = totalParentAnswers/totalQuestions\*100;

$(".progress-bar").css("width", Math.round(progressPercent) + "%");

$(".progress-text").text(Math.round(progressPercent) + "% Completed");

}

*/\*\* Progress Bar END \*/*

*/\*\* State Management \*/*

**function** fetchState() {

**if**(localStorage.getItem("selectedAnswers")!=**undefined**) {

selectedAnswers = JSON.parse(localStorage.getItem("selectedAnswers"));

**if**(selectedAnswers.length>0) {

console.log('setting state back');

clientForms = JSON.parse(localStorage.getItem("clientForms"));

dynamicQuestionOptionJson = JSON.parse(localStorage.getItem("dynamicQuestionOptionJson"));

incident\_id = parseInt(localStorage.getItem("incident\_id"));

currentQuestion = JSON.parse(localStorage.getItem("currentQuestion"));

currentParentKey = parseInt(localStorage.getItem("currentParentKey"));

currentTreeOptions = JSON.parse(localStorage.getItem("currentTreeOptions"));

dynamicQuestionJson = JSON.parse(localStorage.getItem("dynamicQuestionJson"));

currentForm = parseInt(localStorage.getItem("currentForm"));

dependedQuestionsAnswers = JSON.parse(localStorage.getItem("dependedQuestionsAnswers"));

totalQuestions = parseInt(localStorage.getItem("totalQuestions"));

totalParentAnswers = parseInt(localStorage.getItem("totalParentAnswers"));

isThankYouPage = localStorage.getItem('isThankYouPage')=="true"?**true**:**false**;

}

}

*// Set Total Questions Count*

calculateTotalQuestions();

**if**(selectedAnswers.length==0) {

intiateForm();

}

**else** **if**(isThankYouPage) {

onFormSaved(clientForms[currentForm]);

}

**else** {

getDynamicQuestions(currentQuestion.id, **null**);

}

}

**function** saveStateLocally() {

*// Save data to localstorage*

localStorage.setItem("clientForms", JSON.stringify(clientForms));

localStorage.setItem("dynamicQuestionOptionJson", JSON.stringify(dynamicQuestionOptionJson));

localStorage.setItem("incident\_id", incident\_id);

localStorage.setItem("currentQuestion", JSON.stringify(currentQuestion));

localStorage.setItem("currentParentKey", currentParentKey);

localStorage.setItem("currentTreeOptions", JSON.stringify(currentTreeOptions));

localStorage.setItem("selectedAnswers", JSON.stringify(selectedAnswers));

localStorage.setItem("dynamicQuestionJson", JSON.stringify(dynamicQuestionJson));

localStorage.setItem("currentForm", currentForm);

localStorage.setItem("dependedQuestionsAnswers", JSON.stringify(dependedQuestionsAnswers));

localStorage.setItem("totalQuestions", totalQuestions);

localStorage.setItem("totalParentAnswers", totalParentAnswers);

}

**function** resetState() {

localStorage.removeItem("clientForms");

localStorage.removeItem("dynamicQuestionOptionJson");

localStorage.removeItem("incident\_id");

localStorage.removeItem("currentQuestion");

localStorage.removeItem("currentParentKey");

localStorage.removeItem("currentTreeOptions");

localStorage.removeItem("selectedAnswers");

localStorage.removeItem("dynamicQuestionJson");

localStorage.removeItem("currentForm");

localStorage.removeItem("dependedQuestionsAnswers");

localStorage.removeItem("totalQuestions");

localStorage.removeItem("totalParentAnswers");

localStorage.removeItem("isThankYouPage");

}

*/\* State Management End \*/*

**function** showNextForm() {

**if**(clientForms[currentForm+1]!=**undefined**) {

currentForm++;

console.log('moving forward to form '+currentForm);

intiateForm();

} **else** {

console.log("The END!");

}

}

**Safecity End**

**Report Incident**

<?php **if**(!defined('BASEPATH')) **exit**('No direct script access allowed');

*// include\_once APPPATH . '/libraries/BaseController.php';*

*/\*\**

*\* Class : Report\_incident (ReportIncidentController)*

*\* Category Class to control all Category related operations.*

*\* @author : Amita Hadawale*

*\* @version : 1.1*

*\* @since : 22 March 2020*

*\*/*

**class** **Report\_incident** **extends** BaseController

{

*/\*\**

*\* This is default constructor of the class*

*\*/*

**public** **function** \_\_construct()

{

**parent**::\_\_construct();

$this->load->helper('cookie');

$this->load->model('form\_model');

$this->load->model('category\_model');

$this->load->model('incident\_report\_model');

$this->load->model('city\_model');

$this->load->model('country\_model');

*// language changes 29-10-2020 start - sonam*

$lang = (isset($\_COOKIE['language']) && !**empty**($\_COOKIE['language']) ? $\_COOKIE['language'] : 'English');

$this->lang->load('menu', $lang);

$this->lang->load('onboarding', $lang);

$this->lang->load('dynamic\_form', $lang);

$this->lang->load('validation', $lang);

*// language changes 29-10-2020 end - sonam*

}

**public** **function** onboarding()

{

$country\_name = $city\_name = '';

$country\_id = get\_cookie('country\_id')??101; *// Default India*

$city\_id = get\_cookie('city\_id')??1325873; *// Default Mumbai*

$client\_id = 1;

**if**($country\_id!=0) {

$country = $this->country\_model->getById($country\_id);

$country\_name = $country->country\_name??'';

}

**if**($city\_id!=0) {

$city = $this->city\_model->getById($city\_id);

$city\_name = $city->city\_name??'';

$client\_id = $city->client\_id??1;

}

$pageTitle = 'Safecity Webapp';

$this->load->view("onboarding", compact('pageTitle', 'client\_id', 'country\_id', 'country\_name', 'city\_id', 'city\_name'));

}

**public** **function** consent()

{

$this->global['pageTitle'] = 'Safecity Webapp';

$data = $data1 = [];

$data['security\_key'] = '8140c7e293aaa1c933b29b53a2a9140cf176dcfd';

$data['client\_id'] = $\_COOKIE['client\_id'];

$data['lang\_id'] = $\_COOKIE['language\_id'];

$data['country\_id'] = $\_COOKIE['country\_id'];

$data['type'] = 'consent';

$data['content\_for'] = 'web';

$url = base\_url() . 'api/faq/getClientResourceList';

$PData = postAPIData($url,$data);

**if**($PData['er\_msg'] == 'success' || $PData['code'] == 200){

$data1['consent\_data'] = $PData['data'];

$data1['title'] = $PData['title'];

$this->load->view('consent',$data1);

}

}

**public** **function** ngo()

{

$this->global['pageTitle'] = 'Safecity Webapp';

$data = $data1 = [];

$data['security\_key'] = '8140c7e293aaa1c933b29b53a2a9140cf176dcfd';

$data['client\_id'] = $\_COOKIE['client\_id'];

$data['lang\_id'] = $\_COOKIE['language\_id'];

$data['country\_id'] = $\_COOKIE['country\_id'];

$data['ngo\_id'] = $\_COOKIE['ngo\_id'];

$data['type'] = 'ngo';

$data['content\_for'] = 'web';

$url = base\_url() . 'api/ngo/getNgoDetails';

$PData = postAPIData($url,$data);

**if**($PData['er\_msg'] == 'success' || $PData['code'] == 200){

$data1['ngo\_data'] = $PData['data'];

}

$this->load->view('ngo',$data1);

*//}*

}

**public** **function** index()

{

$client\_id = get\_cookie('client\_id')??1;

$lang\_id = get\_cookie('language\_id')??1;

$data\_arr = $this->form\_model->getForms($client\_id, $lang\_id);

$data = ['pageTitle' => 'Safecity Webapp'];

$data['forms'] = json\_encode($data\_arr['forms'], JSON\_UNESCAPED\_UNICODE);

$data['questions'] = json\_encode($data\_arr['questions'], JSON\_UNESCAPED\_UNICODE);

$data['categories'] = json\_encode($data\_arr['categories'], JSON\_UNESCAPED\_UNICODE);

$this->load->view('share\_incident\_form', $data);

}

**public** **function** saveIncident()

{

header('Access-Control-Allow-Origin: \*');

header('Access-Control-Allow-Methods: POST');

*// Load ion auth to check if admin is logged in*

$admin\_id = 0;

$user\_id = $this->input->post('user\_id')??0;

**if**($user\_id==0) {

$this->load->library('ion\_auth');

**if**($this->ion\_auth->logged\_in()) {

$admin\_id = $this->ion\_auth->user()->row()->id;

}

}

$primary\_tags = ['sharing\_for', 'age', 'gender', 'description', 'date', 'time\_from', 'incident\_categories', 'reported\_to\_police', 'attack\_reason', 'additional\_detail', 'incident\_address'];

$answers\_jsons = $this->input->post('answers\_json');

$incident\_id = $this->input->post('incident\_id');

$answers\_jsons = json\_decode($answers\_jsons);

$incident\_arr = [

'is\_public' => 1,

'client\_id' => get\_cookie('client\_id')??1,

'lang\_id' => get\_cookie('language\_id')??1,

'user\_id' => $this->input->post('user\_id')??0,

'admin\_id' => $admin\_id,

'is\_mobile\_visible' => $this->input->post('user\_id')?1:**null**,

'total\_forms' => 1

];

$detail\_arr = [];

**foreach** ($answers\_jsons **as** $answer\_json) {

$question\_answer = $answer\_json->currentQuestion;

$answerJson = $question\_answer->answerJson;

$question\_id = $question\_answer->id;

$question = $question\_answer->question;

$question\_type = json\_decode($question\_answer->properties)->type;

$question\_tags = $question\_answer->tags;

*// incident details data*

$detail\_arr[] = ['incident\_id' => $incident\_id, 'form\_type' => $answerJson->form\_type, 'question\_id' => $question\_id, 'question\_type' => $question\_type, 'question\_tag' => $question\_tags, 'question' => $question, 'answer\_id' => $answerJson->option\_id, 'answer' => $answerJson->answer??'', 'other\_answers' => isset($answerJson->other\_answers)?json\_encode($answerJson->other\_answers):**null**, 'answer\_json' => json\_encode($answerJson)];

*// incident record data*

**if**($incident\_id==0) {

$tags\_arr = explode(',', $question\_tags);

**foreach** ($tags\_arr **as** $tag) {

**if**(in\_array($tag, $primary\_tags)) {

**if**($tag=='date') {

$date\_arr = explode('/', $answerJson->answer);

$incident\_arr[$tag] = $date\_arr[2].'-'.$date\_arr[0].'-'.$date\_arr[1];

$incident\_arr['is\_date\_estimate'] = $answerJson->isEstimate?1:0;

$answerJson->answer = $incident\_arr[$tag];

} **else** **if**($tag=='time\_from') {

$time\_arr = explode('-', $answerJson->answer);

$start\_time = date('H:i:s', strtotime($time\_arr[0]));

$end\_time = count($time\_arr)>1?date('H:i:s', strtotime($time\_arr[1])):**null**;

$incident\_arr['time\_from'] = $start\_time;

$incident\_arr['time\_to'] = $end\_time;

$incident\_arr['is\_time\_estimate'] = $answerJson->isEstimate?1:0;

$answerJson->answer = $incident\_arr['time\_from'].$end\_time!=**null**?'-'.$end\_time:'';

} **else** **if**($tag == 'incident\_address') {

$incident\_arr['building'] = $answerJson->address->building;

$incident\_arr['landmark'] = $answerJson->address->landmark;

$incident\_arr['area'] = $answerJson->address->area;

$incident\_arr['city'] = $answerJson->address->city;

$incident\_arr['state'] = $answerJson->address->state;

$incident\_arr['country'] = $answerJson->address->country;

$incident\_arr['latitude'] = $answerJson->address->latitude;

$incident\_arr['longitude'] = $answerJson->address->longitude;

}

*/\*else if($tag == 'incident\_lat\_lng') {*

*}\*/*

**else** **if**($tag == 'sharing\_for') {

$incident\_arr['sharing\_for'] = $answerJson->option\_id;

}

**else** **if**($tag == 'gender') {

$incident\_arr['gender\_id'] = $answerJson->option\_id;

}

**else** **if**($tag == 'incident\_categories') {

$incident\_arr['incident\_category\_ids'] = $answerJson->option\_id;

}

**else** {

$incident\_arr[$tag] = $answerJson->answer;

}

**break**;

}

}

}

}

$this->db->trans\_start();

*// Save incident*

**if**($incident\_id==0) {

$result\_id = $this->incident\_report\_model->save($incident\_arr);

**if**($result\_id>0) {

$incident\_id = $result\_id;

}

$detail\_arr = array\_map(**function**($detail) **use** ($incident\_id) {

$detail['incident\_id'] = $incident\_id;

**return** $detail;

}, $detail\_arr);

} **else** {

*// Update form count (indicates form type: primary/secondary,etc)*

$this->incident\_report\_model->updateTotalForms($incident\_id);

}

*// Save incident details*

$this->incident\_report\_model->saveDetails($detail\_arr);

$this->db->trans\_complete();

**if** ($this->db->trans\_status() === **FALSE**)

**echo** json\_encode(['succes' => **false**, 'message' => 'Something went wrong!']);

**else**

**echo** json\_encode(['success' => **true**, 'incident\_id' => $incident\_id]);

}

}

**Report incident Api**

<?php **if**(!defined('BASEPATH')) **exit**('No direct script access allowed');

**require\_once**(APPPATH . 'libraries/REST\_Controller.php');

**use** Restserver\Libraries\REST\_Controller;

**class** **Incident\_report** **extends** REST\_Controller

{

*/\*\**

*\* This is default constructor of the class*

*\*/*

**public** **function** \_\_construct()

{

**parent**::\_\_construct();

$this->load->model('report\_incident/incident\_report\_model');

$this->load->model('report\_incident/form\_model');

$this->load->model('report\_incident/question\_model');

$this->load->model('report\_incident/logic\_combination\_model');

$this->load->helper('filter\_helper');

}

*/\*\**

*\* GET reported incidents*

*\* @return JSONArray*

*\*/*

**public** **function** index\_post()

{

*// Validate Request*

**if**($this->post('city')==**null**)

$this->form\_validation->set\_rules('area', 'Area', 'required');

$this->form\_validation->set\_rules('lang\_id', 'Language Id', 'required');

$this->form\_validation->set\_rules('client\_id', 'Client Id', 'required');

**if**(($validator\_arr = runFormValidator($this->form\_validation)) !== **true**)

**return** $this->response($validator\_arr);

*// Retrieve data*

$map\_bounds = $this->post('map\_bound')??[];

**if**(!is\_array($map\_bounds))

$map\_bounds = json\_decode($map\_bounds, **true**);

$map\_zoom = $this->post('map\_zoom')??1;

$city = $this->post('city')??'';

$area = $this->post('area')??'';

$latitude = $this->post('latitude')??'';

$longitude = $this->post('longitude')??'';

$distance = $this->post('distance')??'';

$lang\_id = $this->post('lang\_id')??1;

$client\_id = $this->post('client\_id')??1;

$offset = (int) $this->post('offset')??0;

*// Category Filter*

$cat\_ids = $this->post('categories\_ids');

*// Date Filter*

$reported\_on = $this->post('reported\_on');

$date\_result = calculateDateFilter($reported\_on);

$start\_date = $date\_result['start\_date'];

$end\_date = $date\_result['end\_date'];

*// Time Filter*

$reported\_time\_arr = [];

**if**(trim($this->post('reported\_time'))!='')

$reported\_time\_arr = explode(',', $this->post('reported\_time'));

$time\_arr = [];

**foreach** ($reported\_time\_arr **as** $reported\_time) {

$time\_result = calculateTimeFilter($reported\_time);

$start\_time = $time\_result['start\_time'];

$end\_time = $time\_result['end\_time'];

$time\_arr[] = ['start\_time' => $start\_time, 'end\_time' => $end\_time];

}

**if**($map\_zoom>=15) {

$limit = 10;

*// Get total incident count*

$total\_incidents = $this->incident\_report\_model->getIncidentCounts($map\_bounds, $city, $area, $cat\_ids, $start\_date, $end\_date, $time\_arr, $lang\_id, $client\_id);

*// Get limited incident records*

$incidents = $this->incident\_report\_model->getReports($map\_bounds, $city, $area, $cat\_ids, $start\_date, $end\_date, $time\_arr, $lang\_id, $client\_id, $limit, $offset);

*// Format Records*

$incident\_arr = $this->formatIncidentDetails($incidents);

$current\_count = count($incident\_arr);

} **else** {

$limit = 20;

$incidents = [];

*//$incidents = $this->incident\_report\_model->getReports($map\_bounds, $city, $area, $cat\_ids, $start\_date, $end\_date, $time\_arr, $lang\_id, $client\_id, $limit, $offset);*

$incidents = $this->incident\_report\_model->getReports('', '', '', $cat\_ids, $start\_date, $end\_date, $time\_arr, $lang\_id, $client\_id, $limit, 0);

*// Format Records*

$incident\_arr = $this->formatIncidentDetails($incidents);

$current\_count = count($incident\_arr);

$total\_incidents = $current\_count;

}

*// Mapping Coordinates*

*//$map\_incidents = $this->incident\_report\_model->getIncidentMapCoordinates($map\_bounds, $cat\_ids, $start\_date, $end\_date, $time\_arr, 1);*

$map\_incidents = [];

$this->response([

'status' => **true**,

'message' => 'Incident reports list',

'total' => $total\_incidents,

'limit' => $limit,

'offset' => $offset,

'showing' => ($current\_count==0?0:(($offset+1).' to '.($offset+$current\_count))).' of '.$total\_incidents,

'current\_count' => $current\_count,

'data' => array\_values($incident\_arr),

'map\_data' => $map\_incidents

]);

}

**public** **function** getMapCooordinates\_post()

{

$map\_bounds = $this->post('map\_bound')??[];

**if**(!is\_array($map\_bounds))

$map\_bounds = json\_decode($map\_bounds, **true**);

$client\_id = $this->post('client\_id')??1;

*// Category Filter*

$cat\_ids = $this->post('categories\_ids')??'';

*// Date Filter*

$reported\_on = $this->post('reported\_on');

$date\_result = calculateDateFilter($reported\_on);

$start\_date = $date\_result['start\_date']??'';

$end\_date = $date\_result['end\_date']??'';

*// Time Filter*

$reported\_time\_arr = [];

**if**(trim($this->post('reported\_time'))!='')

$reported\_time\_arr = explode(',', $this->post('reported\_time'));

$time\_arr = [];

**foreach** ($reported\_time\_arr **as** $reported\_time) {

$time\_result = calculateTimeFilter($reported\_time);

$start\_time = $time\_result['start\_time'];

$end\_time = $time\_result['end\_time'];

$time\_arr[] = ['start\_time' => $start\_time, 'end\_time' => $end\_time];

}

$incidents = $this->incident\_report\_model->getIncidentMapCoordinates($map\_bounds, $cat\_ids, $start\_date, $end\_date, $time\_arr, 1);

$this->response([

'status' => **true**,

'message' => 'Incident reports list',

'data' => $incidents

]);

}

**public** **function** update\_post()

{

$incident\_id = $this->input->post('incident\_id');

$incident\_data = json\_decode($this->input->post('incident\_data'), **true**);

$incident\_arr = [

'updated\_on' => date('Y-m-d H:i:s'),

'status' => 'pending\_approval'

];

$primary\_tags = ['sharing\_for', 'age', 'gender', 'description', 'date', 'time\_from', 'incident\_categories', 'reported\_to\_police', 'attack\_reason', 'additional\_detail', 'incident\_address'];

**foreach** ($incident\_data **as** $incident\_detail) {

$question\_tags = $incident\_detail['question\_tag'];

$answer\_id = $incident\_detail['answer\_id'];

$answer = $incident\_detail['answer'];

$answer\_json = json\_decode($incident\_detail['answer\_json']??'{}');

$tags\_arr = explode(',', $question\_tags);

**foreach** ($tags\_arr **as** $tag) {

**if**(in\_array($tag, $primary\_tags)) {

**if**($tag=='date') {

*//$date\_arr = explode('/', $answer);*

*//$incident\_arr[$tag] = $date\_arr[2].'-'.$date\_arr[0].'-'.$date\_arr[1];*

$incident\_arr[$tag] = $answer;

$incident\_arr['is\_date\_estimate'] = $answer\_json->isEstimate?1:0;

$incident\_detail['answer'] = $incident\_arr[$tag];

} **else** **if**($tag=='time\_from') {

$time\_arr = explode('-', $answer);

$start\_time = date('H:i:s', strtotime($time\_arr[0]));

$end\_time = count($time\_arr)>1?date('H:i:s', strtotime($time\_arr[1])):**null**;

$incident\_arr['time\_from'] = $start\_time;

$incident\_arr['time\_to'] = $end\_time;

$incident\_arr['is\_time\_estimate'] = $answer\_json->isEstimate?1:0;

$incident\_detail['answer'] = $incident\_arr['time\_from'].$end\_time!=**null**?'-'.$end\_time:'';

} **else** **if**($tag == 'incident\_address') {

$incident\_arr['building'] = $answer\_json->address->building;

$incident\_arr['landmark'] = $answer\_json->address->landmark;

$incident\_arr['area'] = $answer\_json->address->area;

$incident\_arr['city'] = $answer\_json->address->city;

$incident\_arr['state'] = $answer\_json->address->state;

$incident\_arr['country'] = $answer\_json->address->country;

$incident\_arr['latitude'] = $answer\_json->address->latitude;

$incident\_arr['longitude'] = $answer\_json->address->longitude;

}

**else** **if**($tag == 'sharing\_for') {

$incident\_arr['sharing\_for'] = $answer\_id;

}

**else** **if**($tag == 'gender') {

$incident\_arr['gender\_id'] = $answer\_id;

}

**else** **if**($tag == 'incident\_categories') {

$incident\_arr['incident\_category\_ids'] = $answer\_id;

}

**else** {

$incident\_arr[$tag] = $answer;

}

**break**;

}

}

}

$this->db->trans\_start();

*// Update Incident*

$result = $this->incident\_report\_model->update($incident\_id, $incident\_arr);

*// Update incident details*

$result = $this->incident\_report\_model->updateDetails($incident\_data);

$this->db->trans\_complete();

**if** ($this->db->trans\_status() === **FALSE**)

**return** $this->response([

'status' => **false**,

'message' => 'Something went wrong!',

]);

**else**

**return** $this->response([

'status' => **true**,

'message' => 'Incident updated succesfully',

]);

}

*/\*\**

*\* GET Incident Details*

*\* @return JSONArray*

*\*/*

**public** **function** getDetailedIncidentReport\_post()

{

*// Validate Request*

$this->form\_validation->set\_rules('incident\_id', 'Incident Id', 'required');

**if**(($validator\_arr = runFormValidator($this->form\_validation)) !== **true**)

**return** $this->response($validator\_arr);

$incident\_id = $this->post('incident\_id');

$lang\_id = $this->post('lang\_id')??1;

$incident = $this->incident\_report\_model->getDetailedReport($incident\_id, $lang\_id);

*// Format Records*

$incident\_arr = $this->formatIncidentDetails($incident, **true**);

**if**(count($incident\_arr)==0) {

**return** $this->response([

'status' => **true**,

'message' => 'Incident Details',

'data' => $incident\_arr,

]);

}

$incident\_id = $incident[0]['id'];

$this->response([

'status' => **true**,

'message' => 'Incident Details',

'data' => $incident\_arr[$incident\_id],

]);

}

*/\*\**

*\* GET all reported incidents by a specific users*

*\* @return JSONArray*

*\*/*

**public** **function** getUserIncident\_post()

{

*// Validate Request*

$this->form\_validation->set\_rules('user\_id', 'User Id', 'required');

**if**(($validator\_arr = runFormValidator($this->form\_validation)) !== **true**)

**return** $this->response($validator\_arr);

$user\_id = $this->post('user\_id');

$lang\_id = $this->post('lang\_id')??1;

$incidents = $this->incident\_report\_model->getUserReports($user\_id, $lang\_id);

$incident\_arr = $this->formatIncidentDetails($incidents);

$this->response([

'status' => **true**,

'message' => 'User Incident reports list',

'data' => array\_values($incident\_arr),

]);

}

*/\*\**

*\* DELETE incident report*

*\* @return JSONArray*

*\*/*

**public** **function** deleteUserIncident\_post()

{

*// Validate Request*

$this->form\_validation->set\_rules('user\_id', 'User Id', 'required');

$this->form\_validation->set\_rules('incident\_id', 'Incident Id', 'required');

$this->form\_validation->set\_rules('delete\_from', 'Delete From', 'required');

**if**(($validator\_arr = runFormValidator($this->form\_validation)) !== **true**)

**return** $this->response($validator\_arr);

$user\_id = $this->post('user\_id');

$incident\_id = $this->post('incident\_id');

$delete\_from = $this->post('delete\_from');

**if**($delete\_from=='mobile')

$result = $this->incident\_report\_model->unsetMobileVisibility($incident\_id, $user\_id);

**else**

$result = $this->incident\_report\_model->deleteUserReports($incident\_id, $user\_id);

$this->response([

'status' => $result?**true**:**false**,

'message' => $result?'Incident deleted successfully':'Failed to delete incident',

]);

}

*/\*\**

*\* Helper to merge incident details in incident records itself*

*\* @param Array $incidents*

*\* @return Array*

*\*/*

**private** **function** formatIncidentDetails($incidents, $categorize = **false**)

{

$incident\_arr = [];

**foreach** ($incidents **as** $incident) {

**if**(!isset($incident\_arr[$incident['id']])) {

*// Save incident*

$incident\_arr[$incident['id']] = $incident;

*// Unique categories*

$cat\_ans = explode(',', $incident['categories']);

$incident\_arr[$incident['id']]['categories'] = implode(' | ', array\_unique($cat\_ans));

*// Unset Data not required in main list (duplicates/irrelevant)*

unset($incident\_arr[$incident['id']]['question']);

unset($incident\_arr[$incident['id']]['question\_id']);

unset($incident\_arr[$incident['id']]['question\_type']);

unset($incident\_arr[$incident['id']]['answer\_id']);

unset($incident\_arr[$incident['id']]['form\_type']);

unset($incident\_arr[$incident['id']]['other\_answers']);

unset($incident\_arr[$incident['id']]['question\_tag']);

unset($incident\_arr[$incident['id']]['answer']);

}

*// Save just the answer list*

$unique\_answer = explode(',', $incident['answer']);

$incident\_detail\_list = [

'detail\_id' => $incident['detail\_id'],

'question\_id' => $incident['question\_id'],

'question\_type' => $incident['question\_type'],

'question' => $incident['question'],

'question\_tag' => $incident['question\_tag'],

'answer' => implode(',', array\_unique($unique\_answer)),

'answer\_id' => $incident['answer\_id'],

'answer\_tag' => $incident['answer\_tag'],

'other\_answers' => $incident['other\_answers']

];

**if**($categorize)

$incident\_arr[$incident['id']]['answers'][$incident['form\_type']][$incident['question\_tag']!==''?$incident['question\_tag']:$incident['question\_id']] = $incident\_detail\_list;

**else**

$incident\_arr[$incident['id']]['answers'][] = $incident\_detail\_list;

}

**return** $incident\_arr;

}

*/\*\**

*\* Get All Forms*

*\* @return Array*

*\*/*

**public** **function** getForms\_post()

{

$client\_id = $this->input->post('client\_id')??1;

$lang\_id = $this->input->post('lang\_id')??1;

**if**(!$client\_id || !$lang\_id)

$data\_arr = ['forms' => [], 'questions' => []];

**else**

$data\_arr = $this->form\_model->getForms($client\_id, $lang\_id);

**return** $this->response($data\_arr);

}

*/\*\* Get logical question API \*/*

**public** **function** getLogicalQuestions\_post()

{

$form\_id = $this->input->post('form\_id');

$question\_id = $this->input->post('question\_id');

$ans\_ids = $this->input->post('ans\_ids');

$lang\_id = $this->input->post('lang\_id')??1;

$combinations = $this->logic\_combination\_model->get\_combination\_json($form\_id, $question\_id, $ans\_ids);

**if**(count($combinations)>0) {

$comb\_json\_arr = json\_decode($combinations[0]->comb\_json);

**if**(count($comb\_json\_arr)==0) {

**echo** json\_encode(['comb\_json' => [], 'questions' => []]);

**return** **false**;

}

*// Get unique question ids*

$question\_id\_arr = [];

**foreach** ($comb\_json\_arr **as** $comb\_json\_obj) {

$this->question\_model->getQuestionIdRecursive((**array**) $comb\_json\_obj, $question\_id\_arr);

}

$uniq\_ques\_ids = array\_unique($question\_id\_arr);

*// Get all the questions and the options from the database*

$question\_with\_options = $this->question\_model->get\_questions\_with\_options($uniq\_ques\_ids, $lang\_id);

*// Format the questions as required by application*

$questions = $this->question\_model->formatQuestionOptions($question\_with\_options);

$this->response(['comb\_json' => $comb\_json\_arr, 'questions' => $questions]);

} **else** {

$this->response(['comb\_json' => [], 'questions' => []]);

}

}

**public** **function** saveIncident\_post()

{

*// Load ion auth to check if admin is logged in*

$admin\_id = 0;

$user\_id = $this->input->post('user\_id')??0;

$lang\_id = $this->input->post('lang\_id')??1;

$client\_id = $this->input->post('client\_id')??1;

**if**($user\_id==0) {

$this->load->library('ion\_auth');

**if**($this->ion\_auth->logged\_in()) {

$admin\_id = $this->ion\_auth->user()->row()->id;

}

}

$primary\_tags = ['sharing\_for', 'age', 'gender', 'description', 'date', 'time\_from', 'incident\_categories', 'reported\_to\_police', 'attack\_reason', 'additional\_detail', 'incident\_address'];

$answers\_jsons = $this->input->post('answers\_json');

$incident\_id = $this->input->post('incident\_id');

$answers\_jsons = json\_decode($answers\_jsons);

$incident\_arr = [

'is\_public' => 1,

'client\_id' => $client\_id,

'lang\_id' => $lang\_id,

'user\_id' => $this->input->post('user\_id')??0,

'admin\_id' => $admin\_id,

'is\_mobile\_visible' => $this->input->post('user\_id')?1:**null**,

'total\_forms' => 1

];

$detail\_arr = [];

**foreach** ($answers\_jsons **as** $answer\_json) {

$question\_answer = $answer\_json->currentQuestion;

$answerJson = $question\_answer->answerJson;

$question\_id = $question\_answer->id;

$question = $question\_answer->question;

$question\_type = json\_decode($question\_answer->properties)->type;

$question\_tags = $question\_answer->tags;

*// incident details data*

$detail\_arr[] = ['incident\_id' => $incident\_id, 'form\_type' => $answerJson->form\_type, 'question\_id' => $question\_id, 'question\_type' => $question\_type, 'question\_tag' => $question\_tags, 'question' => $question, 'answer\_id' => $answerJson->option\_id, 'answer' => $answerJson->answer??'', 'other\_answers' => isset($answerJson->other\_answers)?json\_encode($answerJson->other\_answers):**null**, 'answer\_json' => json\_encode($answerJson)];

*// incident record data*

**if**($incident\_id==0) {

$tags\_arr = explode(',', $question\_tags);

**foreach** ($tags\_arr **as** $tag) {

**if**(in\_array($tag, $primary\_tags)) {

**if**($tag=='date') {

$date\_arr = explode('/', $answerJson->answer);

$incident\_arr[$tag] = $date\_arr[2].'-'.$date\_arr[0].'-'.$date\_arr[1];

$incident\_arr['is\_date\_estimate'] = $answerJson->isEstimate?1:0;

$answerJson->answer = $incident\_arr[$tag];

} **else** **if**($tag=='time\_from') {

$time\_arr = explode('-', $answerJson->answer);

$start\_time = date('H:i:s', strtotime($time\_arr[0]));

$end\_time = count($time\_arr)>1?date('H:i:s', strtotime($time\_arr[1])):**null**;

$incident\_arr['time\_from'] = $start\_time;

$incident\_arr['time\_to'] = $end\_time;

$incident\_arr['is\_time\_estimate'] = $answerJson->isEstimate?1:0;

$answerJson->answer = $incident\_arr['time\_from'].$end\_time!=**null**?'-'.$end\_time:'';

} **else** **if**($tag == 'incident\_address') {

$incident\_arr['building'] = $answerJson->address->building;

$incident\_arr['landmark'] = $answerJson->address->landmark;

$incident\_arr['area'] = $answerJson->address->area;

$incident\_arr['city'] = $answerJson->address->city;

$incident\_arr['state'] = $answerJson->address->state;

$incident\_arr['country'] = $answerJson->address->country;

$incident\_arr['latitude'] = $answerJson->address->latitude;

$incident\_arr['longitude'] = $answerJson->address->longitude;

}

*/\*else if($tag == 'incident\_lat\_lng') {*

*}\*/*

**else** **if**($tag == 'sharing\_for') {

$incident\_arr['sharing\_for'] = $answerJson->option\_id;

}

**else** **if**($tag == 'gender') {

$incident\_arr['gender\_id'] = $answerJson->option\_id;

}

**else** **if**($tag == 'incident\_categories') {

$incident\_arr['incident\_category\_ids'] = $answerJson->option\_id;

}

**else** {

$incident\_arr[$tag] = $answerJson->answer;

}

**break**;

}

}

}

}

$this->db->trans\_start();

*// Save incident*

**if**($incident\_id==0) {

$result\_id = $this->incident\_report\_model->save($incident\_arr);

**if**($result\_id>0) {

$incident\_id = $result\_id;

}

$detail\_arr = array\_map(**function**($detail) **use** ($incident\_id) {

$detail['incident\_id'] = $incident\_id;

**return** $detail;

}, $detail\_arr);

} **else** {

*// Update form count (indicates form type: primary/secondary,etc)*

$this->incident\_report\_model->updateTotalForms($incident\_id);

}

*// Save incident details*

$this->incident\_report\_model->saveDetails($detail\_arr);

$this->db->trans\_complete();

**if** ($this->db->trans\_status() === **FALSE**)

**return** $this->response(['succes' => **false**, 'message' => 'Something went wrong!']);

**else**

**return** $this->response(['success' => **true**, 'incident\_id' => $incident\_id]);

}

}

**Safecity\_tips**

<?php **if**(!defined('BASEPATH')) **exit**('No direct script access allowed');

**require\_once**(APPPATH . 'libraries/REST\_Controller.php');

**use** Restserver\Libraries\REST\_Controller;

**class** **Safety\_tips** **extends** REST\_Controller

{

*/\*\**

*\* This is default constructor of the class*

*\*/*

**public** **function** \_\_construct()

{

**parent**::\_\_construct();

$this->load->model('safety\_tips/safety\_tips\_report\_model');

$this->load->helper('filter\_helper');

}

*/\*\**

*\* GET safety tips*

*\* @return JSONArray*

*\*/*

**public** **function** index\_post()

{

*// Validate request*

**if**($this->post('city')==**null**)

$this->form\_validation->set\_rules('area', 'Area', 'required');

$this->form\_validation->set\_rules('lang\_id', 'Language Id', 'required');

*//$this->form\_validation->set\_rules('client\_id', 'Client Id', 'required');*

**if**(($validator\_arr = runFormValidator($this->form\_validation)) !== **true**)

**return** $this->response($validator\_arr);

$map\_bounds = $this->post('map\_bound')??[];

**if**(!is\_array($map\_bounds))

$map\_bounds = json\_decode($map\_bounds, **true**);

$map\_zoom = $this->post('map\_zoom')??1;

$city = $this->post('city')??'';

$area = $this->post('area')??'';

$lang\_id = $this->post('lang\_id')??1;

$client\_id = $this->post('client\_id')??1;

$offset = (int) $this->post('offset')??0;

*// Filters*

$start\_date = $end\_date = '';

$reported\_on = $this->post('reported\_on');

$date\_result = calculateDateFilter($reported\_on);

$start\_date = $date\_result['start\_date']??'';

$end\_date = $date\_result['end\_date']??'';

**if**($map\_zoom>=15) {

*// Get total count*

$total\_safety\_tips = $this->safety\_tips\_report\_model->getSafetyTipsCounts($map\_bounds, $city, $area, $start\_date, $end\_date, $lang\_id, $client\_id);

$limit = 10;

*// Get limited records*

$safety\_tips = $this->safety\_tips\_report\_model->getSafetyTipsByLocation($map\_bounds, $city, $area, $start\_date, $end\_date, $lang\_id, $client\_id, $limit, $offset);

} **else** {

$limit = 20;

*// Get latest 20 records*

$safety\_tips = $this->safety\_tips\_report\_model->getSafetyTipsByLocation('', '', '', $start\_date, $end\_date, $lang\_id, $client\_id, $limit, 0);

$total\_safety\_tips = count($safety\_tips);

}

$current\_count = count($safety\_tips);

$this->response([

'status' => **true**,

'message' => 'Safety tip reports list',

'total' => $total\_safety\_tips,

'limit' => $limit,

'offset' => $offset,

'showing' => ($current\_count==0?0:(($offset+1).' to '.($offset+$current\_count))).' of '.$total\_safety\_tips,

'current\_count' => $current\_count,

'data' => array\_values($safety\_tips),

]);

}

**public** **function** getMapCooordinates\_post()

{

$map\_bounds = $this->post('map\_bound')??[];

**if**(!is\_array($map\_bounds))

$map\_bounds = json\_decode($map\_bounds, **true**);

$client\_id = $this->post('client\_id')??1;

*// Date Filter*

$reported\_on = $this->post('reported\_on');

$date\_result = calculateDateFilter($reported\_on);

$start\_date = $date\_result['start\_date']??'';

$end\_date = $date\_result['end\_date']??'';

$safety\_tips = $this->safety\_tips\_report\_model->getMapCoordinates($map\_bounds, $start\_date, $end\_date, $client\_id);

$this->response([

'status' => **true**,

'message' => 'Safety tips reports list',

'data' => $safety\_tips

]);

}

*/\*\**

*\* GET Safety Tip Details*

*\* @return JSONArray*

*\*/*

**public** **function** getDetailedSafetyTipReport\_post()

{

*// Validate Request*

$this->form\_validation->set\_rules('safety\_tip\_id', 'Safety Tip Id', 'required');

**if**(($validator\_arr = runFormValidator($this->form\_validation)) !== **true**)

**return** $this->response($validator\_arr);

$safety\_tip\_id = $this->post('safety\_tip\_id');

$lang\_id = $this->post('lang\_id')??1;

$safety\_tip = $this->safety\_tips\_report\_model->getDetailedReport($safety\_tip\_id, $lang\_id);

$this->response([

'status' => **true**,

'message' => 'Safety Tip Details',

'data' => $safety\_tip,

]);

}

*/\*\**

*\* UPDATE Safety Tip*

*\* @return JSONArray*

*\*/*

**public** **function** updateSafetyTipReport\_post()

{

*// Validate Request*

$this->form\_validation->set\_rules('safety\_tip\_id', 'Safety Tip Id', 'required');

$this->form\_validation->set\_rules('user\_id', 'User Id', 'required');

*//$this->form\_validation->set\_rules('location\_city\_state', 'City State', 'required');*

*//$this->form\_validation->set\_rules('location', 'location', 'required');*

*//$this->form\_validation->set\_rules('landmark', 'landmark', 'required');*

$this->form\_validation->set\_rules('city', 'city', 'required');

*//$this->form\_validation->set\_rules('state', 'state', 'required');*

$this->form\_validation->set\_rules('country', 'country', 'required');

*//$this->form\_validation->set\_rules('exact\_location', 'Exact location', 'required');*

$this->form\_validation->set\_rules('map\_lat', 'Latitude', 'required');

$this->form\_validation->set\_rules('map\_lon', 'Longitude', 'required');

$this->form\_validation->set\_rules('safety\_tip\_title', 'Title', 'required');

$this->form\_validation->set\_rules('safety\_tip\_desc', 'Description', 'required');

**if**(($validator\_arr = runFormValidator($this->form\_validation)) !== **true**)

**return** $this->response($validator\_arr);

$safety\_tip\_id = $this->post('safety\_tip\_id');

$user\_id = $this->post('user\_id');

$data\_arr = **array**(

'location\_city\_state' => $this->post('location\_city\_state')??'',

'location' => $this->post('location')??'',

'landmark' => $this->post('landmark')??'',

'city' => $this->post('city'),

'state' => $this->post('state')??'',

'country' => $this->post('country'),

'exact\_location' => $this->post('exact\_location')??'',

'map\_lat' => $this->post('map\_lat'),

'map\_lon' => $this->post('map\_lon'),

'safety\_tip\_title' => $this->post('safety\_tip\_title'),

'safety\_tip\_desc' => $this->post('safety\_tip\_desc'),

'updated\_date' => date('Y-m-d H:i:s'),

'status' => 'pending\_approval'

);

$result = $this->safety\_tips\_report\_model->updateReport($safety\_tip\_id, $user\_id, $data\_arr);

$this->response([

'status' => $result?**true**:**false**,

'message' => $result?'Safety Tip updated successfully':'Failed to update safety tip',

]);

}

*/\*\**

*\* GET Safety tips shared by specific mobile user*

*\* @return JSONArray*

*\*/*

**public** **function** getUserSafetyTips\_post()

{

*// Validate Request*

$this->form\_validation->set\_rules('user\_id', 'User Id', 'required');

**if**(($validator\_arr = runFormValidator($this->form\_validation)) !== **true**)

**return** $this->response($validator\_arr);

$user\_id = $this->post('user\_id');

$lang\_id = $this->post('lang\_id')??1;

$safety\_tips = $this->safety\_tips\_report\_model->getUserReports($user\_id, $lang\_id);

$this->response([

'status' => **true**,

'message' => 'User Safety Tip reports list',

'data' => array\_values($safety\_tips),

]);

}