

## 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_ornization"); ?></label>
```

```

custom-radio custom-control-inline">
class="custom-control-input" id="customRadio" name="is_organization" value="1">
control-label" for="customRadio">
<?php echo $this->lang->line("org_yes"); ?>
</label>
</div>
<div class="custom-control
custom-radio custom-control-inline">
class="custom-control-input" id="customRadio2" name="is_organization" value="0">
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>
control js-state" id="autocomplete-organization">
id="organization_id">
</div>
<!-- organization end -->
<!-- language start -->
<!-- <div class="form-group"
id="languageblock">
<label><?php echo $this->lang->line("select_language"); ?></label>
control js-state" id="autocomplete-language" autocomplete="off">
id="language_id">
</div> -->
<div class="form-group"
id="languageblock">
<label><?php echo $this->lang->line("select_language"); ?></label>
language-id" class="form-control js-state">

```

```

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

                        <ul class="progress-form">

                            <div class="Step1">
                                <a id="dynamicBack" href="javascript:void(0);" class="pull-left animated
fadeInUp dynamic_back"><?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> <!-- //questionnaire -->
<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">
    
  </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"></sc
ript>
<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_verification").modal("hide");
        $("#languageblock").show();
        setLanguages();
    } else {
        // Set passcode error message
        $("#varification_code")
            .siblings(".text-danger")
            .html(code_error); //code changes by sonam -
    }
})
.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");
    }
});

```

29-10-2020

```

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_verification").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) {
        initiateForm();
    }
    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);
        initiateForm();
    } 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** getMapCoordinates\_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 getMapCoordinates_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),
    ]);
}

```