# <?PHP?>

# YII 2.0 FRAMEWORK

## USER

### USER IDENTITY

|  |
| --- |
| Yii::$app->user->identity->id |

# ASSSETS

### Register Asset

|  |
| --- |
| backend\assets\AppAsset::register($this); |

### AppAsset.php

|  |
| --- |
| <?php  **namespace** backend\assets;  **use** yii\web\AssetBundle;  /\*\*  \* Main backend application asset bundle.  \*/  **class** AppAsset **extends** AssetBundle  {  **public** $basePath = '@webroot';  **public** $baseUrl = '@web';  **public** $css = [  'css/site.css',  ];  **public** $js = [  ];  /\* public $depends = [  'yii\web\YiiAsset',  'yii\bootstrap\BootstrapAsset',  ]; \*/  } |

### Directory Assets Path

|  |
| --- |
| $directoryAsset = Yii::$app->assetManager->getPublishedUrl('@backend/views/myasset'); |

# RBAC

### List User by Role

|  |
| --- |
| <?= $form->field($model, 'associate\_editor')->dropDownList(  ArrayHelper::map(AuthAssignment::getUsersByAssignment('jeb-associate-editor'),'user.id', 'user.fullname'), ['prompt' => 'Please Select' ]  ) ?> |

## CONTROLLERS

### Get Current Controller / Module

|  |
| --- |
| Yii::$app->controller->id  Yii::$app->controller->module->id |

### Only login user

|  |
| --- |
| **use** yii\filters\AccessControl;  **public** **function** behaviors()  {  **return** [  'access' => [  'class' => AccessControl::className(),  'rules' => [  [  'allow' => **true**,  'roles' => ['@'],  ],  ],  ],  ];  } |

## MODELS

### Required on / when

|  |
| --- |
| [['pro\_name', 'pro\_name\_bi', 'pro\_name\_short', 'pro\_level', 'faculty', 'department', 'status', 'study\_mode'], 'required', 'on' => 'nomenclature'],  [['finalise\_file'], 'required', 'when' => **function**($model){  **return** $model->finalise\_option == 2;  }], |

## VIEWS

|  |
| --- |
|  |

### GridView Index.php

|  |
| --- |
| <?= GridView::widget([  'dataProvider' => $dataProvider,  //'filterModel' => $searchModel,  'columns' => [  ['class' => 'yii\grid\SerialColumn'],  'title:ntext',  [  'attribute' => 'scope\_id',  'value' => **function**($model){  **return** $model->scope->scope\_name;  }    ],  [  'attribute' => 'status',  'format' => 'html',  'filter' => Html::activeDropDownList($searchModel, 'status', $searchModel->journalStatus(),['class'=> 'form-control','prompt' => 'Choose Status']),  'value' => **function**($model){  **return** $model->statusLabel();  }  ]  ['class' => 'yii\grid\ActionColumn',  'contentOptions' => ['style' => 'width: 8.7%'],  'template' => '{view}',  //'visible' => false,  'buttons'=>[  'view'=>**function** ($url, $model) {  **return** '<a href="'.Url::to(['submission/view/', 'id' => $model->id]).'" class="btn btn-warning btn-sm"><span class="glyphicon glyphicon-search"></span> VIEW</a>';  }  ],    ],  ],  ]); ?> |

### Grid View custom width

|  |
| --- |
| <?= GridView::widget([  'dataProvider' => $dataProvider,  'options' => [ 'style' => 'table-layout:fixed;' ],    'columns' => [  ['class' => 'yii\grid\SerialColumn'],    [  'attribute' => 'task\_text',  'format' => 'ntext',  'contentOptions' => [ 'style' => 'width: 80%;' ],  ]  ,  ['class' => 'yii\grid\ActionColumn'],  ],  ]); ?> |

### Grid LinkPager BS4

|  |
| --- |
| <?= GridView::widget([  'dataProvider' => $dataProvider,  'pager' => [  'class' => 'yii\bootstrap4\LinkPager',  ],  'filterModel' => $searchModel, |

### Custom action grid view

|  |
| --- |
| ['class' => 'yii\grid\ActionColumn',  'contentOptions' => ['style' => 'width: 8.7%'],  'template' => '{update}',  //'visible' => false,  'buttons'=>[  'update'=>**function** ($url, $model) {  **return** Html::a('<span class="glyphicon glyphicon-pencil"></span> UPDATE',['question-prime/update/', 'id' => $model->id],['class'=>'btn btn-warning btn-sm']);  }  ],    ],  ['class' => 'yii\grid\ActionColumn',  'contentOptions' => ['style' => 'width: 13%'],  'template' => '{update} {delete}',  //'visible' => false,  'buttons'=>[  'update'=>**function** ($url, $model) {  **return** Html::a('<span class="fas fa-edit"></span>',['article-overwrite/update/', 'id' => $model->id],['class'=>'btn btn-warning btn-sm']);  },  'delete'=>**function** ($url, $model) {  **return** Html::a('<span class="fas fa-trash"></span>', ['delete-article', 'id' => $model->id], [  'class' => 'btn btn-danger btn-sm',  'data' => [  'confirm' => 'Are you sure you want to delete this manuscript?',  'method' => 'post',  ],  ])  ;  }  ],    ],  ---complete  ['class' => 'yii\grid\ActionColumn',  'contentOptions' => ['style' => 'width: 13%'],  'template' => '{view} {update} {delete}',  //'visible' => false,  'buttons'=>[  'update' => **function** ($url, $model) {  **return** Html::a('<span class="fas fa-edit"></span>',['update', 'id' => $model->name],['class'=>'btn btn-warning btn-sm']);  },  'view' => **function** ($url, $model) {  **return** Html::a('<span class="fas fa-eye"></span>',['view', 'id' => $model->name],['class'=>'btn btn-success btn-sm']);  },  'delete' => **function** ($url, $model) {  **return** Html::a('<span class="fas fa-trash"></span>', ['delete-article', 'id' => $model->name], [  'class' => 'btn btn-danger btn-sm',  'data' => [  'confirm' => 'Are you sure you want to delete this manuscript?',  'method' => 'post',  ],  ]) ;  }  ],    ] |

### ListView Widget

|  |
| --- |
| **use** yii\widgets\ListView;  ?>  <div class=**"row"**>  <?= ListView::widget([  'dataProvider' => $dataProvider,  //'filterModel' => $searchModel,  'itemView' => **function**($model){  ?>    <div class=**"col-md-3"**>    <div class=**"text bg-light p-4"**>  <h3><a href=**"#"**><?=$model->course\_code?></a></h3>  <h4><?=$model->course\_name?></h4>  <p>**Kategori:** <?=$model->component->name?></p>  <p><?=**substr**($model->defaultVersion->profile->synopsis,0,100)?>**...**</p>  <p><button type=**"button"** class=**"btn btn-secondary"**>**Sinopsis**</button></p>  </div>  <br />  </div>      <?php  }  ]); ?>  </div> |

### Detail View

|  |
| --- |
| **use** yii\widgets\DetailView;  <style>  **table.detail-view th {**  **width:15%;**  **}**  </style>  <?= DetailView::widget([  'model' => $model,  'attributes' => [  [  'attribute' => 'id',  'label' => 'Manuscript No.',  'value' => **function**($model){  **return** $model->manuscriptNo();  }  ],  'title:ntext',    [  'label' => 'Scope',  'format' => 'html',  'value' => **function**($model){  **return** $model->scope->scope\_name;  }  ],  'abstract:ntext',  'keyword',  [  'attribute' => 'pre\_evaluate\_at',  'label' => 'Submit Time',  'format' => 'datetime'  ],  [  'label' => 'Status',  'format' => 'html',  'value' => **function**($model){  **return** $model->wfLabelBack;  }  ]    ],  ]) ?> |

### Detail view width

|  |
| --- |
| <style>  **table.detail-view th {**  **width:15%;**  **}**  </style> |

### Custom GridView + modal

|  |
| --- |
| <?=$this->render('\_search', ['model' => $searchModel])?>  <style type=**"text/css"**>  **table.padded-table td { padding:10px; font-size:18px; color:#000000; }**  </style>  <?= GridView::widget([  'dataProvider' => $dataProvider,  //'filterModel' => $searchModel,  'layout' => '{items} {pager}',  'showHeader'=> **false**,  'tableOptions' => [  'class' => 'padded-table',  ],  'columns' => [    [  'value' => **function**($model, $key, $index){  **return** $index + 1 . '. ';  }  ],  [  'attribute' =>'course\_code',  'label' => 'Kod Kursus',  'format' => 'raw',  'value' => **function**($model){  **return** ' <a href="#" data-toggle="modal" data-target="#modal-'.$model->id.'">  '.$model->course\_code .'  </a>';  }  ]  ,  //'authKey',  // 'accessToken',  [  'attribute' =>'course\_name',  'label' => 'Nama Kursus',  'format' => 'raw',  'value' => **function**($model){  **return** '    <a href="#" data-toggle="modal" data-target="#modal-'.$model->id.'">  '. **strtoupper**($model->course\_name) .'  </a>  <!-- The Modal -->  <div class="modal fade" id="modal-'.$model->id.'">  <div class="modal-dialog">  <div class="modal-content">    <!-- Modal Header -->  <div class="modal-header">  <h4 class="modal-title">'.$model->course\_code .'</h4>  <button type="button" class="close" data-dismiss="modal">×</button>  </div>    <!-- Modal body -->  <div class="modal-body" >    <div align="right" style="font-size:14px"><a href="#">BM</a> | <a href="#">EN</a></div>    <div style="font-size:16px">  <div class="con-bm">'.$model->defaultVersion->profile->synopsis .'</div>  <div class="con-en" style="display:none">'.$model->defaultVersion->profile->synopsis\_bi .'</div>  </div>  </div>      <div class="modal-footer">  <button type="button" class="btn btn-danger" data-dismiss="modal">Close</button>  </div>    </div>  </div>  </div>    ';    }  ],    ],  ]); ?> |

### GridView Checkbox colum

|  |
| --- |
| <?= GridView::widget([  'dataProvider' => $dataProvider,    'columns' => [  ['class' => 'yii\grid\CheckboxColumn'],  ['class' => 'yii\grid\SerialColumn'],  Controller  **if** ($model->load(Yii::$app->request->post())) {  $post = Yii::$app->request->post();  **if**(**isset**($post['selection'])){  $selection = $post['selection'];  **foreach**($selection **as** $select){  $app = Application::findOne($select); |

### Custom colum

|  |
| --- |
| [  'attribute' => 'status',  'label' => 'status',  'value' => **function**($model){  **if**($model->status == 1){  **return** 'YES';  }**else**{  **return** 'NO';  }    }    ], |

## CRUD – WITH SEARCH

### Controller

|  |
| --- |
| public function actionService($id)  {  $searchModel = new SupplierProvideSearch();  $searchModel->service\_id = $id;  $dataProvider = $searchModel->search(Yii::$app->request->queryParams);  return $this->render('service', [  'searchModel' => $searchModel,  'dataProvider' => $dataProvider,  ]);  } |

### Default Filter

|  |
| --- |
| $searchModel = **new** ProjectSearch();  $params = Yii::$app->request->queryParams;  **if**(!**isset**($params['ProjectSearch'])){  $params['ProjectSearch']['status\_num'] = 20;  }  $dataProvider = $searchModel->search($params);    **return** $this->render('index', [  'searchModel' => $searchModel,  'dataProvider' => $dataProvider,  ]); |

### Default Sort

|  |
| --- |
| $dataProvider = **new** ActiveDataProvider([  'query' => $query,  'sort'=> ['defaultOrder' => ['status'=>**SORT\_ASC**]]  ]); |

### Search single box

|  |
| --- |
| Index.php  <?=$this->render('\_search', ['model' => $searchModel])?>  \_search.php  <?php  **use** yii\helpers\Html;  **use** kartik\widgets\ActiveForm;  /\* @var $this yii\web\View \*/  /\* @var $model backend\models\ClaimSearch \*/  /\* @var $form yii\widgets\ActiveForm \*/  ?>  <?php $form = ActiveForm::begin([  'action' => ['index'],  'method' => 'get',  ]); ?>    <div class=**"row"**>  <div class=**"col-md-8"**></div>  <div class=**"col-md-4"**><?= $form->field($model, 'fasi\_name', ['addon' => ['prepend' => ['content'=>'<span class="glyphicon glyphicon-search"></span>']]])->label(**false**)->textInput(['placeholder' => "Cari Fasilitator..."]) ?></div>  </div>  <?php ActiveForm::**end**(); ?> |

### Search model

|  |
| --- |
| **namespace** frontend\modules\catalog\models;  **use** Yii;  **use** yii\base\Model;  **use** yii\data\ActiveDataProvider;  **use** frontend\models\SupplierProvide;  /\*\*  \* SupplierProvideSearch represents the model behind the search form of `frontend\modules\supplier\models\SupplierProvide`.  \*/  **class** SupplierProvideSearch **extends** SupplierProvide  {    **public** $fullname;    /\*\*  \* @inheritdoc  \*/  **public** **function** rules()  {  **return** [  [['id', 'supplier\_id', 'service\_id'], 'integer'],  [['fullname'], 'safe'],  ];  }  /\*\*  \* @inheritdoc  \*/  **public** **function** scenarios()  {  // bypass scenarios() implementation in the parent class  **return** Model::scenarios();  }  /\*\*  \* Creates data provider instance with search query applied  \*  \* @param array $params  \*  \* @return ActiveDataProvider  \*/  **public** **function** search($params)  {  $query = SupplierProvide::find();    $query->joinWith(['user']);  // add conditions that should always apply here  $dataProvider = **new** ActiveDataProvider([  'query' => $query,  'sort'=> ['defaultOrder' => ['status'=>**SORT\_ASC**]],  'pagination' => [  'pageSize' => 100,  ],  ]);  $this->load($params);  **if** (!$this->validate()) {  // uncomment the following line if you do not want to return any records when validation fails  // $query->where('0=1');  **return** $dataProvider;  }  // grid filtering conditions    $query->andFilterWhere([  'id' => $this->id,  'supplier\_id' => $this->supplier\_id,  'service\_id' => $this->service\_id,  ]);    $dataProvider->**sort**->attributes['fullname'] = [  'asc' => ['user.fullname' => **SORT\_ASC**],  'desc' => ['user.fullname' => **SORT\_DESC**],  ];    $query->andFilterWhere(['like', 'user.fullname', $this->fullname]);      **return** $dataProvider;  }  } |

## Validation

### Unique

|  |
| --- |
| ['username', 'unique', 'targetClass' => '\common\models\User', 'message' => 'This username has already been taken'], |

## FORMS

|  |
| --- |
| **use** yii\helpers\ArrayHelper;  <?= GridView::widget([  'dataProvider' => $dataProvider,  'filterModel' => $searchModel,  'columns' => [  ['class' => 'yii\grid\SerialColumn'],    //'service.service\_name',  [  'attribute' => 'fullname',  'label' => 'Supplier',  'value' => 'user.fullname'  'headerOptions' => ['style'=>'text-align:center'],  'contentOptions' => ['style'=>'text-align:center'],  ],    [  'attribute' => 'company',  'label' => 'Company',  'value' => 'user.supplier.company'  ],    [  'attribute' => 'status',  'label' => 'Status',  'format' => 'html',  'value' => **function**($model, $key, $index, $column){  $color = $model->getWorkflowStatus()->getMetadata('color');  **return** '<span class="label label-'.$color.'">'.$model->getWorkflowStatus()->getLabel().'</span>';  //return $model->getWorkflowStatus()->getLabel();    }  ]      ['class' => 'yii\grid\ActionColumn',  'contentOptions' => ['style' => 'width: 8.7%'],  'template' => '{request}',  'buttons'=>[  'request'=>**function** ($url, $model) {  **return** '<a href="'.Url::to(['/client/request/create', 'service' => $model->service\_id, 'supplier' => $model->supplier\_id]).'" class="btn btn-default btn-sm"><span class="glyphicon glyphicon-bell"></span> Request</a>';  }  ],    ],      ],  ]); ?> |

### Form active form

|  |
| --- |
| **use** yii\widgets\ActiveForm;  <?php $form = ActiveForm::begin(); ?>  <?= $form->field($model, 'fasi\_id')->textInput() ?>  <div class=**"form-group"**>    <?= Html::submitButton('Save', ['class' => 'btn btn-success']) ?>  </div>  <?php ActiveForm::**end**(); ?> |

### Template

|  |
| --- |
| <?= $form->field($model, 'pro\_mode', ['template' => '<div class="row">  <div class="col-sm-3">{label}:<br /><i>(Short)</i> </div>  <div class="col-sm-9">{input}<i>e.g. SAK, SAR, SAB</i>{error}  </div>  </div>'])->textInput() ?>  <?= $form->field($student, 'program', ['template' => '{label}{input}<i>e.g. SAK, SAR, SAB</i>{error}  ']  )->textInput(['maxlength' => 5]) |

### Text input

|  |
| --- |
| <?= $form->field($model, 'fasi\_name')->textInput(['placeholder' => "Cari Fasilitator..."]) ?> |

### Load request model

|  |
| --- |
| **if** ($model->load(Yii::$app->request->post())) {  } |

### Textarea

|  |
| --- |
| <?= $form->field($model, 'address\_postal')->textarea(['rows' => '6']) ?> |

### Form error by form

|  |
| --- |
| <?= $form->errorSummary($model); ?> |

### Form error by model

|  |
| --- |
| **if**($model->getErrors()){  **foreach**($model->getErrors() **as** $error){  **if**($error){  **foreach**($error **as** $e){  Yii::$app->session->addFlash('error', $e);  }  }  }  }  **public** **function** flashError(){  **if**($this->getErrors()){  **foreach**($this->getErrors() **as** $error){  **if**($error){  **foreach**($error **as** $e){  Yii::$app->session->addFlash('error', $e);  }  }  }  }  } |

### Dropdownlist

|  |
| --- |
| **use** yii\helpers\ArrayHelper;  <?= $form->field($model, 'field\_name') ->dropDownList(  ArrayHelper::map(Model::find()->all(),'id', 'show\_value'), ['prompt' => 'Please Select' ]  ) ?>  <?= $form->field($model, 'status')->dropDownList( [1 => 'YES' , 0 => 'NO'] ) ?> |

### Textarea Expandable

|  |
| --- |
| In File Asset  **public** $depends = [  'djabiev\yii\assets\AutosizeTextareaAsset',  ]; |

### Likert

|  |
| --- |
| <table class=**"table table-striped table-hover"**>  <thead>  <tr>  <th>**#**</th>  <th>**Review Items**</th>  <th>**1 = Strongly Disagree**  <br />**5 = Strongly Agree**  <br />**NA = Not Applicable**  </th>  </tr>  </thead>  <tbody>    <?php    $i =1;  **foreach**(ReviewForm::find()->all() **as** $f){  $list= [  1 => '<br />1',  2 => '<br />2',  3 => '<br />3',  4 => '<br />4',  5 => '<br />5',  9 => '<br />NA',  ];  **echo** '<tr>  <td>'.$i.'</td>  <td>'.$f->form\_quest.'</td>  <td>'. $form->field($review, 'q\_'.$i)->radioList($list, ['encode' => **false**, 'separator' => ' &nbsp; &nbsp; ']) ->label(**false**).'</td>  </tr>';  $i++;  }    ?>  </tbody>  </table> |

### Dependent Select

|  |
| --- |
| --view---  <?= $form->field($model, 'course\_id') ->dropDownList(  ArrayHelper::map(Course::find()->orderBy('course\_name ASC')->all(),'id', 'course\_name'), [  'prompt' => 'Please Select',  'onchange'=>'  $.post("'.Yii::$app->urlManager->createUrl('/esiap/program/course-version-list?id=').  '"+$(this).val(),function( data )  {$( "select#programstructure-crs\_version\_id" ).html( data );  });'  ]) ?>  <?= $form->field($model, 'crs\_version\_id')->dropDownList([]) ?>  --controller—  **public** **function** actionCourseVersionList($id){  $list = CourseVersion::find()->where(['course\_id' => $id])->orderBy('created\_at DESC')->all();  **if**($list){  **foreach**($list **as** $ver){  **echo** '<option value="'.$ver->id .'">'.$ver->version\_name .'</option>';  }  }    } |

### Select2

|  |
| --- |
| <?php  **use** kartik\select2\Select2;  **echo** $form->field($model, 'fields')->widget(Select2::classname(), [  'data' => [1=>'satu', 2=>'dua'],  'language' => 'de',  'options' => ['multiple' => **true**,'placeholder' => 'Select a state ...'],  'pluginOptions' => [  'allowClear' => **true**  ],  ])->label('User Fields');  ?>  Bootstrap 4  Manual edit vendor at kartik-v\yii2-krajee-base\src\AssetBundle.php  $lib = 'bootstrap4';  Form  $roles = AuthItem::find()  ->where(['type' => 1])  ->andWhere(['like', 'name', 'jeb-'])  ->orderBy('description ASC')  ->all();    $data = ArrayHelper::map($roles, 'name', 'description');    $model->target\_role = **json\_decode**($model->target\_role);  **echo** $form->field($model, 'target\_role')->widget(Select2::classname(), [  'data' => $data,  'options' => ['multiple' => **true**,'placeholder' => 'Select a target...'],  'pluginOptions' => [  'allowClear' => **true**  ],  ])->label('Targeted Roles');  // Multiple select without model  echo Select2::widget([  'name' => 'state\_2',  'value' => '',  'data' => $data,  'options' => ['multiple' => true, 'placeholder' => 'Select states ...']  ]);  Controller  **if** ($model->load(Yii::$app->request->post())) {  $model->updated\_at = **new** Expression('NOW()');  $model->target\_role = **json\_encode**($model->target\_role);  **if**($model->save()){  Yii::$app->session->addFlash('success', "The email templates has been updated successfully");  **return** $this->redirect('index');  }    } |

### Select2 Ajax Loading

|  |
| --- |
| <http://demos.krajee.com/widget-details/select2#usage-ajax>  ----view----  **use** kartik\select2\Select2;  **use** yii\web\JsExpression;  **use** common\models\User;  <?php  $userDesc = **empty**($model->user\_id) ? '' : User::findOne($model->user\_id)->fullname;  $url = Url::to(['/user/user-list-json']);  **echo** $form->field($model, 'user\_id')->widget(Select2::classname(), [  'initValueText' => $userDesc, // set the initial display text  'options' => ['placeholder' => 'Search for a user ...'],  'pluginOptions' => [  'allowClear' => **true**,  'minimumInputLength' => 3,  'language' => [  'errorLoading' => **new** JsExpression("function () { return 'Waiting for results...'; }"),  ],  'ajax' => [  'url' => $url,  'dataType' => 'json',  'data' => **new** JsExpression('function(params) { return {q:params.term}; }')  ],  'escapeMarkup' => **new** JsExpression('function (markup) { return markup; }'),  'templateResult' => **new** JsExpression('function(user) { return user.text; }'),  'templateSelection' => **new** JsExpression('function (user) { return user.text; }'),  ],  ]);  ?>  ---controller----  **use** yii\db\Query;  **public** **function** actionUserListJson($q = **null**, $id = **null**) {    \Yii::$app->response->format = \yii\web\Response::FORMAT\_JSON;    $out = ['results' => ['id' => '', 'text' => '']];  **if** (!**is\_null**($q)) {  $query = **new** Query;  $query->select('id, fullname AS text')  ->from('user')  ->where(['like', 'fullname', $q])  ->limit(20);  $command = $query->createCommand();  $data = $command->queryAll();  $out['results'] = **array\_values**($data);  }  **elseif** ($id > 0) {  $out['results'] = ['id' => $id, 'text' => User::find($id)->fullname];  }  **return** $out;  } |

### DATEPICKER

|  |
| --- |
| **use** kartik\**date**\DatePicker;  <?=$form->field($model, 'date\_start')->widget(DatePicker::classname(), [  'removeButton' => **false**,  'pluginOptions' => [  'autoclose'=>**true**,  'format' => 'yyyy-mm-dd',  'todayHighlight' => **true**,    ],      ]);  ?> |

### Prepend/append

|  |
| --- |
| **use** kartik\widgets\ActiveForm;  <?= $form->field($model, 'service\_charge', [  'addon' => ['prepend' => ['content'=> Yii::$app->formatter->currencyCode]]  ]); ?>  <?= $form->field($model, 'service\_charge', [  'addon' => ['append' => ['content'=>'%']]  ]); ?> |

### Get label

|  |
| --- |
| <?=$model->getAttributeLabel('fasi\_type\_id')?> |

### Scenario

|  |
| --- |
| $model->scenario = 'signup';  [['username', 'fullname', 'user\_email'], 'required', 'on' => 'signup'], |

### Radio separate place

|  |
| --- |
| <?= $form->field($model, 'work\_part\_time')->radio(['label' => 'Option 1', 'value' => 1, 'uncheck' => **null**]) ?>  <?= $form->field($model, 'work\_part\_time')->radio(['label' => 'Option 2', 'value' => 0, 'uncheck' => **null**]) ?>  Yii will assign a checked or unchecked value to the radio button depending on the value of the stored attribute, so if the value is 0 it will check the button that has the value 0. Your problem seems to have been the hidden input that Yii automatically generates. As others have suggested, you need to set this to null if you want more than one radio button for the same field.  If the user checks another button, then all other radio buttons with the same name will become unchecked. The name of the attribute is generated automatically by Yii when it creates the button. |

### Checkbox

|  |
| --- |
| <td>    <?= $form->field($date, "[{$i}]published")->checkbox(['value' => '1', 'label'=> '']); ?>  </td> |

### Radio list

|  |
| --- |
| **echo** $form->field($model, 'category')->radioList([  1 => 'radio 1',  2 => 'radio 2'  ]);  **echo** $form->field($model, 'evaluate\_option')->radioList($options, ['encode' => **false**, 'separator' => '<br />']) ->label(**false**) ?> |

### Submit button – confirm button

|  |
| --- |
| <?= Html::submitButton('Hantar', ['class' => 'btn btn-primary', 'name' => 'what', 'value' => 'option1', ]) ?> |

### Confirm button

|  |
| --- |
| <?=Html::submitButton('Sokong',  ['class' => 'btn btn-warning', 'name' => 'wfaction', 'value' => 'btn-verify', 'data' => [  'confirm' => 'Pasti Sokong?'  ],  ])?>  <?= Html::a('Delete', ['delete', 'id' => $model->id], [  'class' => 'btn btn-danger',  'data' => [  'confirm' => 'Are you sure you want to delete this item?',  'method' => 'post',  ],  ]) ?> |

### Password repeat rule

|  |
| --- |
| ['password\_repeat', 'compare', 'compareAttribute'=>'password', 'message'=>"Passwords don't match" ], |

### Modal bs3

|  |
| --- |
| **use** yii\bootstrap\Modal;  Modal::begin([  'header' => '<h5>Hello world</h5>',  'toggleButton' => ['label' => 'click me'],  ]);  **echo** 'Say hello...';  Modal::**end**(); |

### Render form

|  |
| --- |
| <?= $this->render('\_form', [  'model' => $model,  'form' => $form,  ]) ?> |

### Validation custom

|  |
| --- |
| **use** yii\base\Model;  **class** MyForm **extends** Model  {  **public** $country;  **public** $token;  **public** **function** rules()  {  **return** [  // an inline validator defined as the model method validateCountry()  ['country', 'validateCountry'],  // an inline validator defined as an anonymous function  ['token', **function** ($attribute, $params, $validator) {  **if** (!**ctype\_alnum**($this->$attribute)) {  $this->addError($attribute, 'The token must contain letters or digits.');  }  }],  ];  }  **public** **function** validateCountry($attribute, $params, $validator)  {  **if** (!**in\_array**($this->$attribute, ['USA', 'Indonesia'])) {  $this->addError($attribute, 'The country must be either "USA" or "Indonesia".');  }  }  } |

### Validation ad hoc

|  |
| --- |
| **public** **function** actionSearch($name, $email)  {  $model = **new** DynamicModel(['name' => $name, 'email' => $email]);  $model->addRule(['name', 'email'], 'string', ['max' => 128])  ->addRule('email', 'email')  ->validate();  **if** ($model->hasErrors()) {  // validation fails  } **else** {  // validation succeeds  }  } |

### Validation Required When ?

|  |
| --- |
| [['finalise\_file'], 'required', 'when' => **function**($model){  **return** $model->finalise\_option == 2;  }], |

### File render

|  |
| --- |
| <?=$this->render('\_finalise\_form', [  'model' => $model,  ]);  ?> |

### Custom Template

|  |
| --- |
| $form->field($cloAs, 'percentage', [  'template' => '{input}{error}',  'options' => [  'tag' => **false**, // Don't wrap with "form-group" div  ]])->textInput()->label(**false**) |

### Validate csrf manual

|  |
| --- |
| **if**(Yii::$app->request->post()){  **if**(Yii::$app->request->validateCsrfToken()){  **echo** 'good';  }  **echo** '<pre>';  **print\_r**(Yii::$app->request->post());  **die**();  } |

### Text Editor TinyMce

|  |
| --- |
| **use** dosamigos\tinymce\TinyMce;  <?= $form->field($model, 'publication\_ethics')->widget(TinyMce::className(), [  'options' => ['rows' => 20],  'language' => 'en',  'clientOptions' => [  'plugins' => [  "advlist autolink lists link charmap print preview anchor",  "searchreplace visualblocks code fullscreen",  "insertdatetime table contextmenu paste"  ],  'toolbar' => "undo redo | styleselect | bold italic | alignleft aligncenter alignright alignjustify | bullist numlist outdent indent | link image"  ]  ]);?>  //simple  <?= $form->field($model, 'para\_text')->widget(TinyMce::className(), [  'options' => ['rows' => 14],  'language' => 'en',  'clientOptions' => [  'plugins' => [  "advlist autolink lists link charmap",  "searchreplace visualblocks code fullscreen",  "paste"  ],  'menubar' => **false**,  'toolbar' => "undo redo | bold italic | alignleft aligncenter alignright alignjustify | bullist numlist outdent indent"  ]  ]);?> |

### Field dropdownlist with others

|  |
| --- |
| **use** richardfan\widget\JSRegister;  $list = ['BIMB'=>'BIMB', 'MAYBANK'=>'MAYBANK', 'CIMB'=>'CIMB', 'BSNB'=>'BSNB', 999 => 'LAIN-LAIN (SILA NYATAKAN)'];  $drop = **false**;  **if**($model->eft\_bank){  **if**(**in\_array**($model->eft\_bank, $list)){  $drop = **true**;  }  }**else**{  $drop = **true**;  }  **if**($drop){  **echo** $form->field($model, 'eft\_bank', ['template' => '{label}<div id="con-bank">{input}</div>{error}'])->dropDownList($list);  }**else**{  **echo** $form->field($model, 'eft\_bank');  }  <?php JSRegister::begin(); ?>  <script>  $**(**"#project-eft\_bank"**).**change**(*function*(){**  ***var*** val **=** $**(*this*).**val**();**  ***if*(**val **==** 999**){**  ***var*** html **=** '<input type="text" id="project-eft\_bank" placeholder="Sila Nyatakan..." class="form-control" name="Project[eft\_bank]" / >'**;**  $**(**"#con-bank"**).**html**(**html**);**  **}**  **});**  </script>  <?php JSRegister::**end**(); ?> |

### Form Create Modal

|  |
| --- |
| **use** yii\bootstrap\Modal;  ---view ---  <p>  <?php **echo** Html::button('Add Course', ['value' => Url::to(['structure-create', 'program' => $model->id]), 'class' => 'btn btn-success', 'id' => 'modalButton']);    Modal::begin([  'header' => '<h4>Add Course</h4>',  'id' =>'modal',  'size' => 'modal-lg'  ]);  **echo** '<div id="modalContent"></div>';  Modal::**end**();  $this->registerJs('  $(function(){  $("#modalButton").click(function(){  $("#modal").modal("show")  .find("#modalContent")  .load($(this).attr("value"));  });  });  ');      ?>      </p>  --- controller ---  **public** **function** actionStructureCreate($program)  {  $model = **new** ProgramStructure();  $model->scenario = 'add-course';  $program = $this->findModel($program);  $pversion = $this->findDevelopmentVersion($program->id);    **if** ($model->load(Yii::$app->request->post())) {  $model->prg\_version\_id = $pversion->id;  **if**($model->save()){  **return** $this->redirect(['structure', 'program' => $program->id]);  }    }  **return** $this->renderAjax('../program-structure/create', [  'model' => $model,  'program' => $program  ]);  } |

### Hidden input

|  |
| --- |
| <?=$form->field($model, 'client\_paid')->hiddenInput(['value' => 1])->label(**false**)?> |

## Formatter

### AsDate, AsTime

|  |
| --- |
| Yii::$app->formatter->asTime($upevent->time\_start, 'php:h:i A')  Yii::$app->formatter->asDate($upevent->date\_start, 'medium') |

### CURRENCY DATE

|  |
| --- |
| 'formatter' => [  'class' => 'yii\i18n\Formatter',  'dateFormat' => 'dd.MM.yyyy',  'decimalSeparator' => '.',  'thousandSeparator' => ', ',  'currencyCode' => 'RM',  'nullDisplay' => '',  ], |

### Get currency

|  |
| --- |
| <?=Yii::$app->formatter->currencyCode?> |

## Query Builder

### Basic

|  |
| --- |
| $rows = (**new** \yii\db\Query())  ->select(['id', 'email'])  ->from('user')  ->where(['last\_name' => 'Smith'])  ->limit(10)  ->all(); |

### find where in array

|  |
| --- |
| $query = ModelName::find()->where(['parentId' => $arr]); |

### Not in array

|  |
| --- |
| $count = Article::find()  ->where(['user\_id' => Yii::$app->user->identity->id])  ->andWhere(['not in', 'status', ['ArticleWorkflow/ra-reject',  'ArticleWorkflow/qa-publish'  ]])  ->**count**(); |

### Like

|  |
| --- |
| $roles = AuthItem::find()  ->where(['type' => 1])  ->andWhere(['like', 'name', 'jeb-'])  ->all(); |

### Not in

|  |
| --- |
| $count = Article::find()->where(['user\_id' => Yii::$app->user->identity->id])->andWhere(['<>', 'status', 'ArticleWorkflow/ra-reject'])->**count**(); |

### Group By

|  |
| --- |
| **public** **static** **function** staffByPosition(){  **return** Staff::find()  ->select('staff\_position.position\_name, COUNT(staff.position\_id) as count\_staff')  ->innerJoin('staff\_postion', 'staff\_position.id = staff.position\_id')  ->where(['staff.staff\_active' => 1])  ->groupBy('staff.position')  ->orderBy('staff\_position.position\_order')  ->all();  } |

### Count

|  |
| --- |
| **return** self::find()  ->where(['semester.is\_current' => 1])  ->**count**(); |

### Update All

|  |
| --- |
| Semester::updateAll(['is\_current' => 0], ['id' => $id]); |

### Condition AND OR grouping

|  |
| --- |
| $query->andFilterWhere(['or',  ['>=', 'YEAR(date\_end)', $this->duration],  ['date\_end' => '0000-00-00']  ]); |

### Between dates

|  |
| --- |
| $sem = self::find()  ->where(['is\_open' => 1])  ->andWhere(['<=', 'open\_at', **new** Expression('NOW()')])  ->andWhere(['>', 'close\_at', **date**('Y-m-d')])  ->one(); |

### Order By - Limit

|  |
| --- |
| **return** FasiEdu::find()  ->select('fasi\_edu.\*')  ->where(['fasi\_id' => $this->id])  ->innerJoin('education\_level', 'education\_level.id = fasi\_edu.level')  ->orderBy('education\_level.level DESC')  ->limit(1)  ->one(); |

### Join

|  |
| --- |
| **return** self::find()  ->leftJoin('semester', 'semester.id = application.semester\_id')  ->where(['semester.is\_current' => 1])  ->**count**();  **return** FasiEdu::find()  ->select('fasi\_edu.\*')  ->where(['fasi\_id' => $this->id])  ->innerJoin('education\_level', 'education\_level.id = fasi\_edu.level')  ->orderBy('education\_level.level DESC')  ->limit(1)  ->one();  $reviewers = UserScope::find()->with([  'user' => **function** ($query) {  $query->select('id, fullname');  }  ])->where(['scope\_id' => $scope, ])->asArray()->all();  **return** self::find()  ->select('sp\_course\_slt\_as.assess\_hour')  ->innerJoin('sp\_course\_slt\_as', 'sp\_course\_assessment.id = sp\_course\_slt\_as.assess\_id AND sp\_course\_slt\_as.crs\_version\_id = sp\_course\_assessment.crs\_version\_id')  ->one(); |

### Model – relationship

|  |
| --- |
| **public** **function** getAssociate(){  **return** $this->hasOne(User::className(), ['id' => 'user\_id']);  }    **public** **function** getUserScopes()  {  **return** $this->hasMany(UserScope::className(), ['user\_id' => 'id']);  }  **public** **function** getCompletedReviewers()  {  **return** $this->hasMany(ArticleReviewer::className(), ['article\_id' => 'id'])->where(['status' => 20]);  } |

### Delete/ Delete All

|  |
| --- |
| $clo\_as = CourseCloAssessment::findOne($id);  $clo\_as->**delete**();  InvoiceItem::deleteAll(['invoice\_id' => $id]);  Customer::deleteAll('status = 3'); |

### Table alias

|  |
| --- |
| $query->joinWith(['application.fasi.user', 'coordinator'=> **function**($q) {  $q->joinWith(['fasi fasi2' => **function**($q) {  $q->joinWith('user user2');  }]);  }]); |

### Db expression now()

|  |
| --- |
| use yii\db\Expression;  $user->created\_at = new Expression('NOW()'); |

## CONFIG

## HTML

### a – anchor

|  |
| --- |
| <?= Html::a('Profile', ['user/view', 'id' => $id], ['class' => 'profile-link']) ?>  The first argument is the title. It's not encoded, so if you're using data entered by the user, you need to encode it with Html::encode(). The second argument is what will be in the href attribute of the <a tag. See [Url::to()](https://www.yiiframework.com/doc/guide/2.0/en/helper-url)for details on what values it accepts. The third argument is an array of tag attributes. |

### Url::to()

|  |
| --- |
| **use** yii\helpers\Url;  // creates a URL to a route: /index.php?r=post%2Findex  **echo** Url::to(['post/index']);  // creates a URL to a route with parameters: /index.php?r=post%2Fview&id=100  **echo** Url::to(['post/view', 'id' => 100]);  // creates an anchored URL: /index.php?r=post%2Fview&id=100#content  **echo** Url::to(['post/view', 'id' => 100, '#' => 'content']);  // creates an absolute URL: http://www.example.com/index.php?r=post%2Findex  **echo** Url::to(['post/index'], **true**);  // creates an absolute URL using the https scheme: https://www.example.com/index.php?r=post%2Findex  **echo** Url::to(['post/index'], 'https');  // currently requested route: /index.php?r=admin%2Fpost%2Findex  **echo** Url::to(['']);  // a relative route with action ID only: /index.php?r=admin%2Fpost%2Findex  **echo** Url::to(['index']);  // a relative route: /index.php?r=admin%2Fpost%2Findex  **echo** Url::to(['post/index']);  // an absolute route: /index.php?r=post%2Findex  **echo** Url::to(['/post/index']);  // using an alias "@posts", which is defined as "/post/index": /index.php?r=post%2Findex  **echo** Url::to(['@posts']); |

## DYNAMIC FORM

### CONTROLLER

|  |
| --- |
| **use** common\models\Model;  **use** yii\helpers\ArrayHelper;  **use** yii\db\Expression;  **public** **function** actionCreate()  {  $model = **new** ConfPaper();  $conf = $this->findConferenceByUrl($confurl);  $model->scenario = 'create';  $authors = [**new** ConfAuthor];    **if** ($model->load(Yii::$app->request->post())) {    $model->updated\_at = **new** Expression('NOW()');  $authors = Model::createMultiple(ConfAuthor::classname());  Model::loadMultiple($authors, Yii::$app->request->post());    **foreach** ($authors **as** $i => $author) {  $author->author\_order = $i;  }      $valid = $model->validate();    $valid = Model::validateMultiple($authors) && $valid;    **if** ($valid) {  $transaction = Yii::$app->db->beginTransaction();  **try** {  **if** ($flag = $model->save(**false**)) {  **foreach** ($authors **as** $i => $author) {  **if** ($flag === **false**) {  **break**;  }  //do not validate this in model  $author->paper\_id = $model->id;  **if** (!($flag = $author->save(**false**))) {  **break**;  }  }  }  **if** ($flag) {  $transaction->commit();  Yii::$app->session->addFlash('success', "New Paper is successfully created");  **return** $this->redirect(['update', 'confurl'=> $confurl, 'id' => $model->id]);  } **else** {  $transaction->rollBack();  }  } **catch** (Exception $e) {  $transaction->rollBack();    }  }        }    **return** $this->render('create', [  'model' => $model,  'authors' => (**empty**($authors)) ? [**new** ConfAuthor] : $authors  ]);      }  **public** **function** actionUpdate()  {  $model = $this->findModel($conf);  $dates = $model->confDates;    **if** ($model->load(Yii::$app->request->post())) {    $model->updated\_at = **new** Expression('NOW()');    $oldIDs = ArrayHelper::map($dates, 'id', 'id');      $dates = Model::createMultiple(ConfDate::classname(), $dates);    Model::loadMultiple($dates, Yii::$app->request->post());    $deletedIDs = **array\_diff**($oldIDs, **array\_filter**(ArrayHelper::map($dates, 'id', 'id')));    **foreach** ($dates **as** $i => $date) {  $date->date\_order = $i;  }      $valid = $model->validate();    $valid = Model::validateMultiple($dates) && $valid;    **if** ($valid) {  $transaction = Yii::$app->db->beginTransaction();  **try** {  **if** ($flag = $model->save(**false**)) {  **if** (! **empty**($deletedIDs)) {  ConfDate::deleteAll(['id' => $deletedIDs]);  }  **foreach** ($dates **as** $i => $date) {  **if** ($flag === **false**) {  **break**;  }  //do not validate this in model  $date->conf\_id = $model->id;  **if** (!($flag = $date->save(**false**))) {  **break**;  }  }  }  **if** ($flag) {  $transaction->commit();  Yii::$app->session->addFlash('success', "Dates updated");  **return** $this->redirect(['dates','conf' => $conf]);  } **else** {  $transaction->rollBack();  }  } **catch** (Exception $e) {  $transaction->rollBack();    }  }        }    **return** $this->render('dates', [  'model' => $model,  'dates' => (**empty**($dates)) ? [**new** ConfDate] : $dates  ]);    } |

### VIEW

|  |
| --- |
| <?php  **use** yii\helpers\Html;  **use** yii\widgets\ActiveForm;  **use** wbraganca\dynamicform\DynamicFormWidget;  **use** yii\jui\JuiAsset;  **use** kartik\**date**\DatePicker;  /\* @var $this yii\web\View \*/  /\* @var $model backend\modules\conference\models\Conference \*/  $this->title = 'IMPORTANT DATES: ' . $model->conf\_abbr;  $this->params['breadcrumbs'][] = ['label' => 'Conferences', 'url' => ['index']];  $this->params['breadcrumbs'][] = ['label' => $model->id, 'url' => ['view', 'id' => $model->id]];  $this->params['breadcrumbs'][] = 'Update';  ?>  <div class=**"conference-update"**>  <div class=**"card"**>  <div class=**"card-body"**>  <?php $form = ActiveForm::begin(['id' => 'dynamic-form']); ?>  <?=$form->field($model, 'updated\_at')->hiddenInput(['value' => **time**()])->label(**false**)?>  <?php DynamicFormWidget::begin([  'widgetContainer' => 'dynamicform\_wrapper',  'widgetBody' => '.container-items',  'widgetItem' => '.date-item',  'limit' => 10,  'min' => 1,  'insertButton' => '.add-date',  'deleteButton' => '.remove-date',  'model' => $dates[0],  'formId' => 'dynamic-form',  'formFields' => [  'id',  'date\_name',  'date\_start'  ],  ]); ?>    <table class=**"table table-bordered table-striped"**>  <thead>  <tr>  <th width=**"5%"**></th>  <th>**Date Label**</th>  <th width=**"25%"**>**Date**</th>  <th class=**"text-center"** style=**"width: 90px;"**>    </th>  </tr>  </thead>  <tbody class=**"container-items"**>  <?php **foreach** ($dates **as** $i => $date): ?>  <tr class=**"date-item"**>  <td class=**"sortable-handle text-center vcenter"** style=**"cursor: move;"**>  <i class=**"fas fa-arrows-alt"**></i>  </td>    <td class=**"vcenter"**>  <?php  // necessary for update action.  **if** (! $date->isNewRecord) {  **echo** Html::activeHiddenInput($date, "[{$i}]id");  }  ?>  <?= $form->field($date, "[{$i}]date\_name")->label(**false**) ?>  </td>    <td class=**"vcenter"**>      <?=$form->field($date, "[{$i}]date\_start")->widget(DatePicker::classname(), [  'removeButton' => **false**,  'pluginOptions' => [  'autoclose'=>**true**,  'format' => 'yyyy-mm-dd',  'todayHighlight' => **true**,    ],      ])->label(**false**);  ?>  </td>  <td class=**"text-center vcenter"** style=**"width: 90px; verti"**>  <button type=**"button"** class=**"remove-date btn btn-default btn-sm"**><span class=**"fa fa-remove"**></span></button>  </td>  </tr>  <?php **endforeach**; ?>  </tbody>    <tfoot>  <tr>  <td></td>  <td colspan=**"2"**>  <button type=**"button"** class=**"add-date btn btn-default btn-sm"**><span class=**"fa fa-plus"**></span> **New Dates**</button>    </td>  <td>      </td>  </tr>  </tfoot>    </table>  <?php DynamicFormWidget::**end**(); ?>    <br />  <div class=**"form-group"**>  <?= Html::submitButton('Save Dates', ['class' => 'btn btn-primary']) ?>  </div>  <?php ActiveForm::**end**(); ?>        </div>  </div>  </div>  <?php  $js = <<<'EOD'  jQuery(".dynamicform\_wrapper").on("afterInsert", function(e, item) {  $( ".krajee-datepicker" ).each(function() {  $(this).removeData().kvDatepicker('destroy');  $(this).kvDatepicker(eval($(this).attr('data-krajee-kvdatepicker')));  });  });  var fixHelperSortable = function(e, ui) {  ui.children().each(function() {  $(this).width($(this).width());  });  return ui;  };  $(".container-items").sortable({  items: "tr",  cursor: "move",  opacity: 0.6,  axis: "y",  handle: ".sortable-handle",  helper: fixHelperSortable,  update: function(ev){  $(".dynamicform\_wrapper").yiiDynamicForm("updateContainer");  }  }).disableSelection();  EOD;  JuiAsset::register($this);  $this->registerJs($js);  ?> |

### MODEL FILE

|  |
| --- |
| <?php  **namespace** common\models;  **use** Yii;  **use** yii\helpers\ArrayHelper;  **class** Model **extends** \yii\base\Model  {  /\*\*  \* Creates and populates a set of models.  \*  \* @param string $modelClass  \* @param array $multipleModels  \* @return array  \*/  **public** **static** **function** createMultiple($modelClass, $multipleModels = [])  {  $model = **new** $modelClass;  $formName = $model->formName();  $post = Yii::$app->request->post($formName);  $models = [];  **if** (! **empty**($multipleModels)) {  $keys = **array\_keys**(ArrayHelper::map($multipleModels, 'id', 'id'));  $multipleModels = **array\_combine**($keys, $multipleModels);  }  **if** ($post && **is\_array**($post)) {  **foreach** ($post **as** $i => $item) {  **if** (**isset**($item['id']) && !**empty**($item['id']) && **isset**($multipleModels[$item['id']])) {  $models[] = $multipleModels[$item['id']];  } **else** {  $models[] = **new** $modelClass;  }  }  }  **unset**($model, $formName, $post);  **return** $models;  }  } |

## NeSTED - CONTROLLER

|  |
| --- |
| **protected** **function** findPerson($id)  {  **if** (($model = Person::findOne($id)) !== **null**) {  **return** $model;  } **else** {  **throw** **new** NotFoundHttpException('The requested page does not exist.');  }  }    **public** **function** actionTest($id){  $modelPerson = $this->findPerson($id);  $modelsHouse = $modelPerson->houses;  $modelsRoom = [];  $oldRooms = [];  **if** (!**empty**($modelsHouse)) {  **foreach** ($modelsHouse **as** $indexHouse => $modelHouse) {  $rooms = $modelHouse->rooms;  $modelsRoom[$indexHouse] = $rooms;  $oldRooms = ArrayHelper::merge(ArrayHelper::index($rooms, 'id'), $oldRooms);  }  }  **if** ($modelPerson->load(Yii::$app->request->post())) {    //echo '<pre>';  //print\_r(Yii::$app->request->post());die();  // reset  $modelsRoom = [];  $oldHouseIDs = ArrayHelper::map($modelsHouse, 'id', 'id');  $modelsHouse = Model::createMultiple(House::classname(), $modelsHouse);  Model::loadMultiple($modelsHouse, Yii::$app->request->post());  $deletedHouseIDs = **array\_diff**($oldHouseIDs, **array\_filter**(ArrayHelper::map($modelsHouse, 'id', 'id')));  // validate person and houses models  $valid = $modelPerson->validate();  $valid = Model::validateMultiple($modelsHouse) && $valid;  $roomsIDs = [];  //print\_r( $\_POST['Room']);die();  **if** (**isset**($\_POST['Room'][0][0])) {  **foreach** ($\_POST['Room'] **as** $indexHouse => $rooms) {    $roomsIDs = ArrayHelper::merge($roomsIDs, **array\_filter**(ArrayHelper::getColumn($rooms, 'id')));  //print\_r(ArrayHelper::getColumn($rooms, 'id'));die();  **foreach** ($rooms **as** $indexRoom => $room) {    $data['Room'] = $room;  $modelRoom = (**isset**($room['id']) && **isset**($oldRooms[$room['id']])) ? $oldRooms[$room['id']] : **new** Room;  **echo** '<pre>';**print\_r**($modelRoom);**echo** '<br /><br /><br /><br />';  $modelRoom->load($data);  //print\_r($modelRoom);die();  $modelsRoom[$indexHouse][$indexRoom] = $modelRoom;  $valid = $modelRoom->validate();  }  }  }  $oldRoomsIDs = ArrayHelper::getColumn($oldRooms, 'id');  $deletedRoomsIDs = **array\_diff**($oldRoomsIDs, $roomsIDs);  **if** ($valid) {  $transaction = Yii::$app->db->beginTransaction();  **try** {  **if** ($flag = $modelPerson->save(**false**)) {  **if** (! **empty**($deletedRoomsIDs)) {  Room::deleteAll(['id' => $deletedRoomsIDs]);  }  **if** (! **empty**($deletedHouseIDs)) {  House::deleteAll(['id' => $deletedHouseIDs]);  }  **foreach** ($modelsHouse **as** $indexHouse => $modelHouse) {  **if** ($flag === **false**) {  **break**;  }  $modelHouse->person\_id = $modelPerson->id;  **if** (!($flag = $modelHouse->save(**false**))) {  **break**;  }  **if** (**isset**($modelsRoom[$indexHouse]) && **is\_array**($modelsRoom[$indexHouse])) {    **foreach** ($modelsRoom[$indexHouse] **as** $indexRoom => $modelRoom) {  //echo '<pre>';print\_r($modelRoom);die();  $modelRoom->house\_id =  $modelHouse->id;  **if** (!($flag = $modelRoom->save(**false**))) {  **break**;  }  }  }  }  }  **if** ($flag) {  $transaction->commit();  Yii::$app->session->addFlash('success', "Data Updated");  **return** $this->redirect(['test', 'id' => $modelPerson->id]);  } **else** {  $transaction->rollBack();  }  } **catch** (Exception $e) {  $transaction->rollBack();  }  }  }      **return** $this->render('test', [  'modelPerson' => $modelPerson,  'modelsHouse' => (**empty**($modelsHouse)) ? [**new** House] : $modelsHouse,  'modelsRoom' => (**empty**($modelsRoom)) ? [[**new** Room]] : $modelsRoom  ]);  } |

## Nested – View

|  |
| --- |
| <?php  **use** yii\helpers\Html;  **use** yii\bootstrap\ActiveForm;  **use** wbraganca\dynamicform\DynamicFormWidget;  ?>  <div class=**"person-form"**>  <?php $form = ActiveForm::begin(['id' => 'dynamic-form']); ?>  <div class=**"row"**>  <div class=**"col-sm-6"**>  <?= $form->field($modelPerson, 'first\_name')->textInput(['maxlength' => **true**]) ?>  </div>  <div class=**"col-sm-6"**>  <?= $form->field($modelPerson, 'last\_name')->textInput(['maxlength' => **true**]) ?>  </div>  </div>  <div class=**"padding-v-md"**>  <div class=**"line line-dashed"**></div>  </div>  <?php DynamicFormWidget::begin([  'widgetContainer' => 'dynamicform\_wrapper',  'widgetBody' => '.container-items',  'widgetItem' => '.house-item',  'limit' => 10,  'min' => 1,  'insertButton' => '.add-house',  'deleteButton' => '.remove-house',  'model' => $modelsHouse[0],  'formId' => 'dynamic-form',  'formFields' => [  'description',  ],  ]); ?>  <table class=**"table table-bordered table-striped"**>  <thead>  <tr>  <th>**Houses**</th>  <th style=**"width: 450px;"**>**Rooms**</th>  <th class=**"text-center"** style=**"width: 90px;"**>  <button type=**"button"** class=**"add-house btn btn-success btn-xs"**><span class=**"fa fa-plus"**></span></button>  </th>  </tr>  </thead>  <tbody class=**"container-items"**>  <?php **foreach** ($modelsHouse **as** $indexHouse => $modelHouse): ?>  <tr class=**"house-item"**>  <td class=**"vcenter"**>  <?php  // necessary for update action.  **if** (! $modelHouse->isNewRecord) {  **echo** Html::activeHiddenInput($modelHouse, "[{$indexHouse}]id");  }  ?>  <?= $form->field($modelHouse, "[{$indexHouse}]description")->label(**false**)->textInput(['maxlength' => **true**]) ?>  </td>  <td>  <?= $this->render('\_form-rooms', [  'form' => $form,  'indexHouse' => $indexHouse,  'modelsRoom' => $modelsRoom[$indexHouse],  ]) ?>  </td>  <td class=**"text-center vcenter"** style=**"width: 90px; verti"**>  <button type=**"button"** class=**"remove-house btn btn-danger btn-xs"**><span class=**"fa fa-minus"**></span></button>  </td>  </tr>  <?php **endforeach**; ?>  </tbody>  </table>  <?php DynamicFormWidget::**end**(); ?>    <div class=**"form-group"**>  <?= Html::submitButton($modelPerson->isNewRecord ? 'Create' : 'Update', ['class' => 'btn btn-primary']) ?>  </div>  <?php ActiveForm::**end**(); ?>  </div>  -----  <?php  **use** yii\helpers\Html;  **use** wbraganca\dynamicform\DynamicFormWidget;  //print\_r($modelsRoom);die();  ?>  <?php DynamicFormWidget::begin([  'widgetContainer' => 'dynamicform\_inner',  'widgetBody' => '.container-rooms',  'widgetItem' => '.room-item',  'limit' => 4,  'min' => 1,  'insertButton' => '.add-room',  'deleteButton' => '.remove-room',  'model' => $modelsRoom[0],  'formId' => 'dynamic-form',  'formFields' => [  'description'  ],  ]); ?>  <table class=**"table table-bordered"**>  <thead>  <tr>  <th>**Description**</th>  <th class=**"text-center"**>  <button type=**"button"** class=**"add-room btn btn-success btn-xs"**><span class=**"glyphicon glyphicon-plus"**></span></button>  </th>  </tr>  </thead>  <tbody class=**"container-rooms"**>  <?php **foreach** ($modelsRoom **as** $indexRoom => $modelRoom): ?>  <tr class=**"room-item"**>  <td class=**"vcenter"**>  <?php  // necessary for update action.  **if** (! $modelRoom->isNewRecord) {  **echo** Html::activeHiddenInput($modelRoom, "[{$indexHouse}][{$indexRoom}]id");  }  ?>  <?= $form->field($modelRoom, "[{$indexHouse}][{$indexRoom}]description")->label(**false**)->textInput(['maxlength' => **true**]) ?>  </td>  <td class=**"text-center vcenter"** style=**"width: 90px;"**>  <button type=**"button"** class=**"remove-room btn btn-danger btn-xs"**><span class=**"glyphicon glyphicon-minus"**></span></button>  </td>  </tr>  <?php **endforeach**; ?>  </tbody>  </table>  <?php DynamicFormWidget::**end**(); ?> |

## Nested – sql

|  |
| --- |
| --  -- Table structure for table `house`  --  **CREATE** **TABLE** `house` **(**  `id` **int(**11**)** **NOT** **NULL,**  `person\_id` **int(**11**)** **NOT** **NULL,**  `description` **varchar(**255**)** **NOT** **NULL**  **)** ENGINE**=**InnoDB **DEFAULT** CHARSET**=**latin1**;**  --  -- Dumping data for table `house`  --  **INSERT** **INTO** `house` **(**`id`**,** `person\_id`**,** `description`**)** **VALUES**  **(**10**,** 1**,** 'rumah 1'**),**  **(**11**,** 1**,** 'Rumah'**);**  -- --------------------------------------------------------  --  -- Table structure for table `person`  --  **CREATE** **TABLE** `person` **(**  `id` **int(**11**)** **NOT** **NULL,**  `first\_name` **varchar(**200**)** **NOT** **NULL,**  `last\_name` **varchar(**200**)** **NOT** **NULL**  **)** ENGINE**=**InnoDB **DEFAULT** CHARSET**=**latin1**;**  --  -- Dumping data for table `person`  --  **INSERT** **INTO** `person` **(**`id`**,** `first\_name`**,** `last\_name`**)** **VALUES**  **(**1**,** 'sajer'**,** 'test'**);**  -- --------------------------------------------------------  --  -- Table structure for table `room`  --  **CREATE** **TABLE** `room` **(**  `id` **int(**11**)** **NOT** **NULL,**  `house\_id` **int(**11**)** **NOT** **NULL,**  `description` **varchar(**255**)** **NOT** **NULL**  **)** ENGINE**=**InnoDB **DEFAULT** CHARSET**=**latin1**;**  --  -- Dumping data for table `room`  --  **INSERT** **INTO** `room` **(**`id`**,** `house\_id`**,** `description`**)** **VALUES**  **(**2**,** 10**,** 'bilik satu'**),**  **(**3**,** 10**,** 'fff'**),**  **(**4**,** 11**,** 'Bilik Kosong'**);**  --  -- Indexes for dumped tables  --  --  -- Indexes for table `house`  --  **ALTER** **TABLE** `house`  **ADD** **PRIMARY** **KEY** **(**`id`**),**  **ADD** **KEY** `person\_id` **(**`person\_id`**);**  --  -- Indexes for table `person`  --  **ALTER** **TABLE** `person`  **ADD** **PRIMARY** **KEY** **(**`id`**);**  --  -- Indexes for table `room`  --  **ALTER** **TABLE** `room`  **ADD** **PRIMARY** **KEY** **(**`id`**),**  **ADD** **KEY** `house\_id` **(**`house\_id`**);**  --  -- AUTO\_INCREMENT for dumped tables  --  --  -- AUTO\_INCREMENT for table `house`  --  **ALTER** **TABLE** `house`  **MODIFY** `id` **int(**11**)** **NOT** **NULL** AUTO\_INCREMENT**,** AUTO\_INCREMENT=1**;**  --  -- AUTO\_INCREMENT for table `person`  --  **ALTER** **TABLE** `person`  **MODIFY** `id` **int(**11**)** **NOT** **NULL** AUTO\_INCREMENT**,** AUTO\_INCREMENT**=**1**;**  --  -- AUTO\_INCREMENT for table `room`  --  **ALTER** **TABLE** `room`  **MODIFY** `id` **int(**11**)** **NOT** **NULL** AUTO\_INCREMENT**,** AUTO\_INCREMENT**=**1**;**  --  -- Constraints for dumped tables  --  --  -- Constraints for table `house`  --  **ALTER** **TABLE** `house`  **ADD** **CONSTRAINT** `house\_ibfk\_1` **FOREIGN** **KEY** **(**`person\_id`**)** **REFERENCES** `person` **(**`id`**);** |

## Manual dynamic form

|  |
| --- |
| $tag = Yii::$app->request->post('tagged\_staff');  **if**($tag){  $kira\_post = **count**($tag);  $kira\_lama = **count**($model->pubTagsNotMe);  **if**($kira\_post > $kira\_lama){  $bil = $kira\_post - $kira\_lama;  **for**($i=1;$i<=$bil;$i++){  $insert = **new** PubTag;  $insert->pub\_id = $model->id;  $insert->save();  }  }**else** **if**($kira\_post < $kira\_lama){    $bil = $kira\_lama - $kira\_post;  $deleted = PubTag::find()  ->where(['pub\_id'=>$model->id])  ->andwhere(['<>', 'staff\_id', Yii::$app->user->identity->staff->id])  ->limit($bil)  ->all();  **if**($deleted){  **foreach**($deleted **as** $del){  $del->**delete**();  }  }  }    $update\_tag = PubTag::find()  ->where(['pub\_id' => $model->id])  ->andWhere(['<>', 'staff\_id', Yii::$app->user->identity->staff->id])  ->all();    **if**($update\_tag){  $i=0;  **foreach**($update\_tag **as** $ut){  $ut->staff\_id = $tag[$i];  $ut->save();  $i++;  }  }  }  } |

## UPLOAD FILE – LIMION

Views – main.php upload folder template

Upload.php in common models

### VIEW

|  |
| --- |
| **use** common\models\UploadFile;  $model->file\_controller = 'document';  <?=UploadFile::fileInput($model, 'submission')?>  <!-- for image -->  <?=UploadFile::fileInput($model, 'banner', **true**)?> |

### MODEL

|  |
| --- |
| **Additional property**  **public** $submission\_instance;  **public** $file\_controller;  **Rules**  [['submission\_file'], 'required', 'on' => 'submission\_upload'],  [['submission\_instance'], 'file', 'skipOnEmpty' => **true**, 'extensions' => 'doc, docx', 'maxSize' => 5000000],  [['updated\_at'], 'required', 'on' => 'submission\_delete'], |

### Alias

|  |
| --- |
| In common config – one off setting only  'aliases' => [  '@upload' => '@frontend/uploaded',  '@img' => '@frontend/uploaded/images',  ], |

### controller

|  |
| --- |
| **use** common\models\UploadFile;  **use** yii\helpers\Json;  **use** yii\db\Expression;  **public** **function** actionUploadFile($attr, $id){  $attr = $this->clean($attr);  $model = $this->findModel($id);  $model->file\_controller = 'submission';  **return** UploadFile::upload($model, $attr, 'document\_updated\_at');  }  **protected** **function** clean($string){  $allowed = ['pubupload'];    **foreach**($allowed **as** $a){  **if**($string == $a){  **return** $a;  }  }    **throw** **new** NotFoundHttpException('Invalid Attribute');  }  **public** **function** actionDeleteFile($attr, $id)  {  $attr = $this->clean($attr);  $model = $this->findModel($id);  $attr\_db = $attr . '\_file';    $file = Yii::getAlias('@upload/' . $model->{$attr\_db});    $model->scenario = $attr . '\_delete';  $model->{$attr\_db} = '';  $model->updated\_at = **new** Expression('NOW()');  **if**($model->save()){  **if** (**is\_file**($file)) {  **unlink**($file);    }    **return** Json::encode([  'good' => 1,  ]);  }**else**{  **return** Json::encode([  'errors' => $model->getErrors(),  ]);  }    }  **public** **function** actionDownloadFile($attr, $id, $identity = **true**){  $attr = $this->clean($attr);  $model = $this->findModel($id);  $filename = **strtoupper**($attr) . ' ' . Yii::$app->user->identity->fullname;        UploadFile::download($model, $attr, $filename);  } |

## WORKFLOW

GITHUB: <https://github.com/raoul2000/yii2-workflow>

### COMPOSER

php composer.phar require --prefer-dist raoul2000/yii2-workflow "\*"

"raoul2000/yii2-workflow": "\*"

|  |
| --- |
| Here is the PHP class that implements the definition for our workflow :  @app/models/PostWorkflow.php  namespace app\models;  class PostWorkflow implements \raoul2000\workflow\source\file\IWorkflowDefinitionProvider  {  public function getDefinition() {  return [  'initialStatusId' => 'draft',  'status' => [  'draft' => [  'transition' => ['publish','deleted']  ],  'publish' => [  'transition' => ['draft','deleted']  ],  'deleted' => [  'transition' => ['draft']  ]  ]  ];  }  } |

|  |
| --- |
| @app/models/Post.php  namespace app\models;  /\*\*  \* @property integer $id  \* @property string $title  \* @property string $body  \* @property string $status column used to store the status of the post  \*/  class Post extends \yii\db\ActiveRecord  {  public function behaviors()  {  return [  \raoul2000\workflow\base\SimpleWorkflowBehavior::className()  ];  } |

### Workflow definition

|  |
| --- |
| public function getDefinition() {  return [  'initialStatusId' => 'client-request',  'status' => [  'client-request' => [  'label' => 'Client Request',  'transition' => ['supplier-quotation', 'client-request'],  'metadata' => [  'color' => 'green',  //'icon' => 'fa fa-bell'  ]  ],  'supplier-quotation' => [  'label' => 'Supplier Quotation',  'transition' => ['client-accept'],  'metadata' => [  'color' => 'yellow',  //'icon' => 'fa fa-bell'  ]  ],  'client-accept' => [  'label' => 'Client Acceptance',  'transition' => ['client-payment'],  'metadata' => [  'color' => 'grey',  //'icon' => 'fa fa-bell'  ]  ],  'client-payment' => [  'label' => 'Client Payment',  'transition' => ['supplier-operation'],  'metadata' => [  'color' => 'grey',  //'icon' => 'fa fa-bell'  ]  ],  'supplier-operation' => [  'label' => 'Supplier Operation',  'transition' => ['supplier-delivery'],  'metadata' => [  'color' => 'blue',  //'icon' => 'fa fa-bell'  ]  ],  'supplier-delivery' => [  'label' => 'Supplier Delivery',  'transition' => ['client-review'],  'metadata' => [  'color' => 'black',  //'icon' => 'fa fa-bell'  ]  ],  'client-review' => [  'label' => 'Client Review',  'transition' => ['supplier-payment', 'supplier-operation'],  'metadata' => [  'color' => 'green',  //'icon' => 'fa fa-bell'  ]  ],  'supplier-payment'  ]  ]  ;  } |

### Class loader if not app/models

e.g. common/models

|  |
| --- |
| $config = [  'components' => [  'workflowSource' => [  'class' => 'raoul2000\workflow\source\file\WorkflowFileSource',  'definitionLoader' => [  'class' => 'raoul2000\workflow\source\file\PhpClassLoader',  'namespace' => 'common/models/workflows'  ]  ], |

### Gridview

|  |
| --- |
| [  'attribute' => 'status',  'label' => 'Status',  'value' => **function**($model){  //$text = '<span class="label label-info">Processing</span>';  **return** $model->getWorkflowStatus()->getLabel();  }  ] |

### Change Status

|  |
| --- |
| $model->scenario = WorkflowScenario::enterStatus('ca-assign-reviewer');  $model->asgn\_reviewer\_at = **new** Expression('NOW()');  $model->sendToStatus('ca-assign-reviewer');  **if**($model->save()){  Yii::$app->session->addFlash('success', "Associate Editor has been successfully assigned.");  **return** $this->redirect('index');  } |

### Guide Link

[yii2-workflow Usage Guide](http://raoul2000.github.io/yii2-workflow/)

## CHART

### Polar/Radar

|  |
| --- |
| **use** dosamigos\chartjs\ChartJs;  <?php  **echo** ChartJs::widget([  'type' => 'radar',  'id' => 'main\_dimension',  'options' => [  'height' => 100,  'width' => 100,    /\* 'layout' => [  'padding' => [  'left' => 100,  'right' => 0,  'top' => 0,  'bottom' => 0  ]  ] \*/      ],  'data' => [    'labels' => ['INNOVATION CAPABILITY', 'INNOVATION COMPETENCY', 'INNOVATION OUTPUT',], // Your labels  'datasets' => [  [  'data' => [3,4,5], // Your dataset  'label' => 'Company',  'backgroundColor' => 'rgba(255, 99, 132, 0.2)',  'borderColor' => 'rgba(200,0,0,0.6)',  'fill' => **true**,  'borderWidth' => 3,  'radius' => 6,  'pointStyle' => 'circle',  'pointRadius' => 6,  'pointBorderWidth' => 3,  'pointBackgroundColor' => "orange",  'pointBorderColor' => "rgba(200,0,0,0.6)",  'pointHoverRadius' => 10,  ],  [  'data' => [1,2,3], // Your dataset  'label' => 'Benchmark',  'backgroundColor' => 'rgba(0 , 189, 102, 0.2)',  'borderColor' => 'rgba(0,199,9,0.6)',  'fill' => **true**,  'borderWidth' => 3,  'radius' => 6,  'pointStyle' => 'circle',  'pointRadius' => 6,  'pointBorderWidth' => 3,  'pointBackgroundColor' => "orange",  'pointBorderColor' => "rgba(200,0,0,0.6)",  'pointHoverRadius' => 10,  ]  ]  ],  'clientOptions' => [  'scale' => [  'gridLines' => [  'circular' => **true**,  ],  'ticks' => [  'beginAtZero' => **false**,  'min' => 1,  'max' => 5,  'stepSize' => 0.5,      ],      ],  /\* 'legend' => false \*/  'legend' => [  'position' => 'top',  'labels' => [  'margin' => [  'bottom' => 0  ]  ]  ],        ],    ])  ?> |

## EMAIL

### Extension

"nterms/yii2-mailqueue": "\*",

### Config

|  |
| --- |
| 'mailer' => [  'class' => 'yii\swiftmailer\Mailer',  'viewPath' => '@common/mail',  // send all mails to a file by default. You have to set  // 'useFileTransport' to false and configure a transport  // for the mailer to send real emails.  'useFileTransport' => **false**,  'transport' => [  'class' => 'Swift\_SmtpTransport',  'host' => 'smtp.gmail.com', // e.g. smtp.mandrillapp.com or smtp.gmail.com  'username' => 'pusatko.efasi@gmail.com',  'password' => 'pusatko123',  'port' => '465', // Port 25 is a very common port too  'encryption' => 'ssl', // It is often used, check your provider or mail server specs  ],  ],    'mailqueue' => [  'class' => 'nterms\mailqueue\MailQueue',  'table' => '{{%mail\_queue}}',  'mailsPerRound' => 10,  'maxAttempts' => 3,  'transport' => [  'class' => 'Swift\_SmtpTransport',  'host' => 'smtp.gmail.com',  'username' => 'pusatko.efasi@gmail.com',  'password' => 'pusatko123',  'port' => '465',  'encryption' => 'ssl',  ],  ], |

### Test

|  |
| --- |
| **public** **function** actionTestEmail($token){  **if**($token=='1qaz'){  $user = Yii::$app->user->identity;  $name = $user->fullname;  $email = $user->email;  $send = Yii::$app->mailqueue->compose(  ['html' => 'release-email-html',  'text' => 'release-email-text'],  ['name' => $name]  )  ->setFrom('pusatko.efasi@gmail.com')  ->setTo($email)  ->setSubject('Surat Tawaran Perlantikan Fasilitator')  ->queue();  }    } |

### Example queue

|  |
| --- |
| **private** **function** sendEmail($app){  $user = $app->fasi->user;  $name = $user->fullname;  $email = $user->email;  **if**($app->release\_email == 0){  $send = Yii::$app->mailqueue->compose(  ['html' => 'release-email-html',  'text' => 'release-email-text'],  ['name' => $name]  )  ->setFrom('pusatko.efasi@gmail.com')  ->setTo($email)  ->setSubject('Surat Tawaran Perlantikan Fasilitator')  ->queue();    **if**($send){  $app->scenario = 'release\_email';  $app->release\_email = 1;  $app->save();  }  }  } |

## Admin Login as User

### Table

|  |
| --- |
| CREATE TABLE `user\_token` (  `id` int(11) NOT NULL,  `user\_id` int(11) NOT NULL,  `token` varchar(255) NOT NULL,  `created\_at` int(11) NOT NULL  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;  --  -- Indexes for dumped tables  --  --  -- Indexes for table `user\_token`  --  ALTER TABLE `user\_token`  ADD PRIMARY KEY (`id`); |

### Frontend Site Controller

|  |
| --- |
| **public** **function** actionAdminLogin($id, $token)  {  **if** (!Yii::$app->user->isGuest) {  Yii::$app->user->logout();  }    $last5 = **time**() - (60);    $db = UserToken::find()  ->where(['user\_id' => $id, 'token' => $token])  ->andWhere('created\_at > ' . $last5)  ->one();    **if**($db){  $user = User::findIdentity($id);  **if**(Yii::$app->user->login($user)){  **return** $this->goHome();  }  }**else**{  **throw** **new** ForbiddenHttpException;  }        }  **public** **function** behaviors()  {  **return** [  'access' => [  'class' => AccessControl::className(),  'rules' => [  [  'actions' => ['signup', 'index', 'login', 'admin-login'],  'allow' => **true**,  'roles' => ['?'],  ],  [  'actions' => ['logout', 'index', 'admin-login'],  'allow' => **true**,  'roles' => ['@'],  ],  ],  ],  'verbs' => [  'class' => VerbFilter::className(),  'actions' => [  'logout' => ['post'],  ],  ],  ];  } |

### Button

|  |
| --- |
| **return** '<a href="'.Url::to(['customer/login-as/', 'id' => $model->id]).'" class="btn btn-'.$color.' btn-sm" target="\_blank"><span class="glyphicon glyphicon-user"></span> Login As</a>'; |

### Backend Controller

|  |
| --- |
| **public** **function** actionLoginAs($id)  {  $token = **new** UserToken;  $token->setToken();  $user = Yii::$app->user->identity->id;  **if**($token->save()){  **return** $this->redirect(Yii::$app->urlManager->createUrl('./../site/admin-login?id='.$id.'&token='.$token->token));  }  } |

### UserToken model – Just copy

## TEMPLATE BOX/CARD/CONTAINER

### ADMINLTE Container

|  |
| --- |
| <div class=**"box"**>  <div class=**"box-header"**></div>  <div class=**"box-body"**>  **---- content here --------------**  </div>  </div> |

### ADMINLTE Container With Title

|  |
| --- |
| <div class=**"box"**>  <div class=**"box-header"**>  <h3 class=**"box-title"**>  **Title Text Here**  </h3>  </div>  <div class=**"box-body"**>  **------content here--------**  </div>  </div> |

### FKP FRONTEND

|  |
| --- |
| <section class=**"contact-page spad pt-0"**>  <div class=**"container"**>    **----content here ----**    </div>  </section> |

## BOOTSTRAP

### Grid

|  |
| --- |
| <div class=**"row"**>  <div class=**"col-md-6"**></div>  <div class=**"col-md-6"**></div>  </div> |

### Card

|  |
| --- |
| <div class=**"card"**>  <div class=**"card-header"**>**Members**</div>  <div class=**"card-body"**>**Content**</div>  <div class=**"card-footer"**>**Footer**</div>  </div> |

## Kartik-v

|  |
| --- |
|  |

## ChartJs

### Pie Chart

|  |
| --- |
| **use** dosamigos\chartjs\ChartJs;  **use** yii\helpers\ArrayHelper;  $result = StaffStats::staffByType();  $data = ArrayHelper::map($result, 'is\_academic','count\_staff');  $data = **array\_values**($data);  //print\_r($data);  **echo** ChartJs::widget([  'type' => 'pie',  'id' => 'structurePie',  'options' => [  'height' => 200,  'width' => 400,  ],  'data' => [  'radius' => "90%",  'labels' => ['Academic', 'Administrative'], // Your labels  'datasets' => [  [  'data' => $data, // Your dataset  'label' => '',  'backgroundColor' => [  '#ADC3FF',  '#FF9A9A',  'rgba(190, 124, 145, 0.8)'  ],  'borderColor' => [  '#fff',  '#fff',  '#fff'  ],  'borderWidth' => 1,  'hoverBorderColor'=>["#999","#999","#999"],  ]  ]  ],  'clientOptions' => [  'legend' => [  'display' => **true**,  'position' => 'bottom',  'labels' => [  'fontSize' => 10,  'fontColor' => "#425062",  ]  ],  'tooltips' => [  'enabled' => **true**,  'intersect' => **true**  ],  'hover' => [  'mode' => **false**  ],  'maintainAspectRatio' => **false**,  ],  'plugins' =>  **new** \yii\web\JsExpression("  [{  afterDatasetsDraw: function(chart, easing) {  var ctx = chart.ctx;  chart.data.datasets.forEach(function (dataset, i) {  var meta = chart.getDatasetMeta(i);  if (!meta.hidden) {  meta.data.forEach(function(element, index) {  // Draw the text in black, with the specified font  ctx.fillStyle = 'rgb(0, 0, 0)';  var fontSize = 13;  var fontStyle = 'normal';  var fontFamily = 'Helvetica';  ctx.font = Chart.helpers.fontString(fontSize, fontStyle, fontFamily);  // Just naively convert to string for now  var dataString = dataset.data[index].toString();  // Make sure alignment settings are correct  ctx.textAlign = 'center';  ctx.textBaseline = 'middle';  var padding = 5;  var position = element.tooltipPosition();  ctx.fillText(dataString, position.x, position.y - (fontSize / 2) - padding);  });  }  });  }  }]")  ])  ?> |

### Bar Horizontal

|  |
| --- |
| **use** dosamigos\chartjs\ChartJs;  **use** yii\helpers\ArrayHelper;  $position\_result = StaffStats::staffByPosition();  $position = ArrayHelper::map($position\_result, 'id','position\_name');  $position\_data = ArrayHelper::map($position\_result, 'id','count\_staff');  $position = **array\_values**($position);  $position\_data = **array\_values**($position\_data);  **echo** ChartJs::widget([  'type' => 'horizontalBar',  'options' => [    'width' => 400,  'height' => 400  ],  'data' => [  'labels' => $position,  'datasets' => [  [  'label' => "Total Staff",  'backgroundColor' => '#327fa8',  'borderColor' => "rgba(179,181,198,1)",  'pointBackgroundColor' => "rgba(179,181,198,1)",  'pointBorderColor' => "#fff",  'pointHoverBackgroundColor' => "#fff",  'pointHoverBorderColor' => "rgba(179,181,198,1)",  'data' => $position\_data  ],  ]  ]  ]);  ?> |

## Database

### Transaction

|  |
| --- |
| $transaction = Yii::$app->db->beginTransaction();  **try** {  $transaction->commit();    }  **catch** (Exception $e)  {  $transaction->rollBack();  Yii::$app->session->addFlash('error', $e->getMessage());  } |

## OTHERS

### Export

|  |
| --- |
| **use** kartik\grid\GridView;  **use** kartik\export\ExportMenu;  $exportColumns = [  ['class' => 'yii\grid\SerialColumn'],  'course\_code',  'course\_name',  'course\_name\_bi',  'credit\_hour',  [  'attribute' => 'program.pro\_name\_short',  'label' => 'Program',  ],  ];  <?=ExportMenu::widget([  'dataProvider' => $dataProvider,  'columns' => $exportColumns,  'filename' => 'COURSE\_DATA\_' . **date**('Y-m-d'),  'onRenderSheet'=>**function**($sheet, $grid){  $sheet->getStyle('A2:'.$sheet->getHighestColumn().$sheet->getHighestRow())  ->getAlignment()->setWrapText(**true**);  },  'exportConfig' => [  ExportMenu::FORMAT\_PDF => **false**,  ExportMenu::FORMAT\_EXCEL\_X => **false**,  ],  ]);?>  <div class=**"box"**>  <div class=**"box-header"**></div>  <div class=**"box-body"**><?= GridView::widget([  'dataProvider' => $dataProvider,  'export' => **false**,  // 'filterModel' => $searchModel,  'columns' => [  ['class' => 'yii\grid\SerialColumn'],  [  'attribute' => 'course\_code',  'contentOptions' => ['style' => 'width: 10%'],    ],  ... |

### Language

|  |
| --- |
| $lang = Yii::$app->language; |

### Template

|  |
| --- |
| <div class=**"block-content"**>  <div class=**"container"**>    <div class=**"row"**>  <div class=**"col"**>  <h2 class=**"section\_title text-center"**>**EDITORIAL COMMITTEES**</h2>  </div>  </div>  </div>  </div> |

### Get Params Config

|  |
| --- |
| Yii::$app->params['faculty\_id']  Yii::$app->params['faculty\_abbr']  'faculty\_id' => Yii::$app->params['faculty\_id'] |

### Get params url

|  |
| --- |
| Yii::$app->getRequest()->getQueryParam('companyAdd\_id') |

##### **Sarjana Muda Keusahawanan** **(Keusahawanan Kesihatan) Dengan Kepujian**

### Register JS-rechardfan

|  |
| --- |
| **use** richardfan\widget\JSRegister;  <?php JSRegister::begin(['position' => **static**::POS\_BEGIN]); ?>  <script>  ***function*** newPopup**(**url**)** **{**  popupWindow **=** window.open**(**url**,**'popUpWindow'**,**'height=300,width=400,left=10,top=10,resizable=yes,scrollbars=yes,toolbar=yes,menubar=no,location=no,directories=no,status=yes'**)**  **}**  </script>  <?php JSRegister::**end**(); ?> |

### Register JS

|  |
| --- |
| <?php  $js = "  $('#element ').click(function(e, data){  alert('register');  });  $('#element ').change(function(e, data){  alert('register');  });  ";  $this->registerJs($js);  ?>  <?php  $js = <<<'EOD'  EOD;  $this->registerJs($js);  ?> |

### Session

|  |
| --- |
| $session = Yii::$app->session;  // get a session variable. The following usages are equivalent:  $language = $session->get('language');  $language = $session['language'];  $language = **isset**($\_SESSION['language']) ? $\_SESSION['language'] : **null**;  // set a session variable. The following usages are equivalent:  $session->set('language', 'en-US');  $session['language'] = 'en-US';  $\_SESSION['language'] = 'en-US';  // remove a session variable. The following usages are equivalent:  $session->remove('language');  **unset**($session['language']);  **unset**($\_SESSION['language']);  // check if a session variable exists. The following usages are equivalent:  **if** ($session->has('language')) ...  **if** (**isset**($session['language'])) ...  **if** (**isset**($\_SESSION['language'])) ...  // traverse all session variables. The following usages are equivalent:  **foreach** ($session **as** $name => $value) ...  **foreach** ($\_SESSION **as** $name => $value) ... |

### Flash message

|  |
| --- |
| Yii::$app->session->addFlash('error', "Data failed!");  Yii::$app->session->addFlash('success', "Data Updated"); |

Recipient

### Javascript redirect

|  |
| --- |
| // similar behavior as an HTTP redirect  window.location.replace("http://stackoverflow.com");  // similar behavior as clicking on a link  window.location.href = "http://stackoverflow.com"; |

### Https

|  |
| --- |
| RewriteEngine On  RewriteCond %{HTTPS} off  RewriteRule ^ https://%{HTTP\_HOST}%{REQUEST\_URI} [L,NE,R=301]  RewriteCond %{REQUEST\_FILENAME} !-f  RewriteCond %{REQUEST\_FILENAME} !-d  RewriteRule ^(.\*)\?\*$ index.php?r=$1 [L,QSA] |

### Random String

|  |
| --- |
| Yii::$app->security->generateRandomString() |

### Cron Job

|  |
| --- |
| /usr/local/bin/php /home/pusatkoumkedu/efasi/yii cron/send >/dev/null 2>&1 |

### Datetime format

|  |
| --- |
| **date**('F d<\s\up>S</\s\up>, Y', $this->model->invoice\_ts)  <?php // Prints the day echo date("l") . "<br>";  // Prints the day, date, month, year, time, AM or PM echo date("l jS \of F Y h:i:s A"); ?>  Specifies the format of the outputted date string. The following characters can be used:  d - The day of the month (from 01 to 31)  D - A textual representation of a day (three letters)  j - The day of the month without leading zeros (1 to 31)  l (lowercase 'L') - A full textual representation of a day  N - The ISO-8601 numeric representation of a day (1 for Monday, 7 for Sunday)  S - The English ordinal suffix for the day of the month (2 characters st, nd, rd or th. Works well with j)  w - A numeric representation of the day (0 for Sunday, 6 for Saturday)  z - The day of the year (from 0 through 365)  W - The ISO-8601 week number of year (weeks starting on Monday)  F - A full textual representation of a month (January through December)  m - A numeric representation of a month (from 01 to 12)  M - A short textual representation of a month (three letters)  n - A numeric representation of a month, without leading zeros (1 to 12)  t - The number of days in the given month  L - Whether it's a leap year (1 if it is a leap year, 0 otherwise)  o - The ISO-8601 year number  Y - A four digit representation of a year  y - A two digit representation of a year  a - Lowercase am or pm  A - Uppercase AM or PM  B - Swatch Internet time (000 to 999)  g - 12-hour format of an hour (1 to 12)  G - 24-hour format of an hour (0 to 23)  h - 12-hour format of an hour (01 to 12)  H - 24-hour format of an hour (00 to 23)  i - Minutes with leading zeros (00 to 59)  s - Seconds, with leading zeros (00 to 59)  u - Microseconds (added in PHP 5.2.2)  e - The timezone identifier (Examples: UTC, GMT, Atlantic/Azores)  I (capital i) - Whether the date is in daylights savings time (1 if Daylight Savings Time, 0 otherwise)  O - Difference to Greenwich time (GMT) in hours (Example: +0100)  P - Difference to Greenwich time (GMT) in hours:minutes (added in PHP 5.1.3)  T - Timezone abbreviations (Examples: EST, MDT)  Z - Timezone offset in seconds. The offset for timezones west of UTC is negative (-43200 to 50400)  c - The ISO-8601 date (e.g. 2013-05-05T16:34:42+00:00)  r - The RFC 2822 formatted date (e.g. Fri, 12 Apr 2013 12:01:05 +0200)  U - The seconds since the Unix Epoch (January 1 1970 00:00:00 GMT) |