$model->file = UploadedFile::getInstance($model, 'file'); if ($model->upload()) {

return $this->renderContent("File {$model->file->name} is uploaded

successfully");

}

}

return $this->render('index', ['model' => $model]);

}

}

3. Finally, you can view @app/views/upload/index. php as follows:

<?php

use yii\widgets\ActiveForm; use yii\helpers\Html;

?>

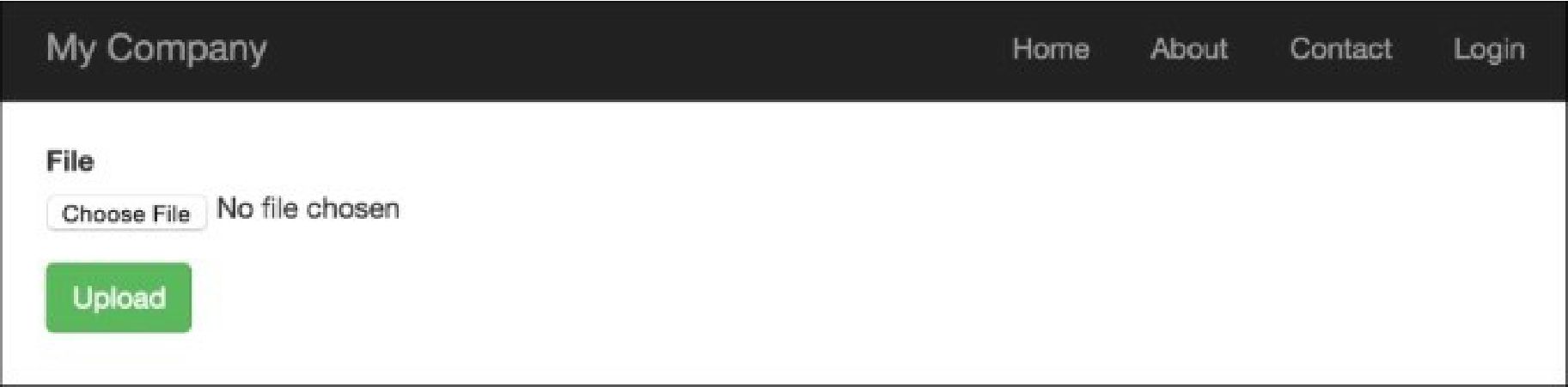
<?php $form = ActiveForm::begin(['options' => ['enctype' => 'multipart/form- data']]) ?>

<?= $form->field($model, 'file')->fileInput() ?>

<?= Html::submitButton('Upload', ['class' => 'btn-success'])?>

<?php ActiveForm::end() ?>

4. That is it. Now, run the upload controller and try uploading both ZIP archives and other files, as shown in the following screenshot:



How it works...

The model we use is pretty simple. We define only one field, named $file, and a validation rule that uses the FileValidator file validator, which reads only ZIP files.

We create a model instance and fill it with data from $\_post if the form is submitted:

$model->file = UploadedFile::getInstance($model, 'file'); if ($model->upload()) {

return $this->renderContent("File {$model->file->name} is uploaded successfully");

}

We then use UploadedFile: : getInstance, which gives us access to use the UploadedFile instance. This class is a wrapper around the $\_file array that PHP fills when the file is uploaded. We make sure that the file is a ZIP archive by calling the model’s validate method, then we save the file using UploadedFile::saveAs.