

```

-- Corresponds to file ....PHP Files/write.php
module Handler.Write where

import Import
import Data.ByteString.Lazy as B

getWriteR :: Handler Html
getWriteR = do
    liftIO $ B.writeFile "/Users/Nickolas/Documents/write.txt" "The quick brown fox jumps over the lazy dog."
    defaultLayout [whamlet|Written|]

-- Corresponds to file ....PHP Files/register.php
module Handler.Register where

import Import
import Yesod.Form.Jquery
import Prelude
instance YesodJquery App

personForm :: Form Person
personForm = renderDivs $ Person
    <$> areq textField "Name" Nothing
    <*> areq (jqueryDayField def
        {jdsChangeYear = True
        , jdsYearRange = "1900:-5"
        }) "Birthday" Nothing
    <*> aopt textField "Favorite color" Nothing
    <*> areq emailField "Email" Nothing
    <*> aopt urlField "Website" Nothing

getRegisterR :: Handler Html
getRegisterR = do
    (widget, enctype) <- generateFormPost personForm
    defaultLayout
        [whamlet|
            <p>
                Enter your information to register!
                <form method=post action=@{RegisterR} enctype=#{enctype}>
                    ^{widget}
                    <button>Submit
            |]

postRegisterR :: Handler Html
postRegisterR = do
    ((result, widget), enctype) <- runFormPost personForm
    case result of
        FormSuccess person -> do
            personKey <- runDB $ insert $ person
            defaultLayout [whamlet|Success! Here is your key: #{show person}|]
        _ -> do
            -- This is a test case because Yesod doesn't like to give FormSuccess when we request the page 10,000 times.
            -- It will always execute when we're just using the Adobe Benchmarking tool.
            user <- runDB $ insert $ Person "george" (read "2015-04-11") (Just "blue") "email" (Just "website")
            defaultLayout
                [whamlet|Success!|]

-- Corresponds to file ....PHP Files/read.php
module Handler.Read where

import Import
import qualified Data.ByteString as B

getReadR :: Handler Html
getReadR = do
    contents <- liftIO $ B.readFile "/Users/Nickolas/Documents/write.txt"
    defaultLayout [whamlet|Contents of text.txt: #{show contents}|]

-- Corresponds to file ....PHP Files/database.php
module Handler.Database where

import Import

getDatabaseR :: Handler Html
getDatabaseR = do
    people <- runDB $ selectList ([] :: [Filter Person]) []
    defaultLayout [whamlet|Person: #{show people}|]

```