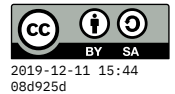


# Assignment 11



due *Tue 10 Dec*

For this assignment, use Warp and Blaze HTML to create a small web site that responds to four types of pages:

1. / : the home page
2. /about : an about page
3. /user/ID : a user profile page, where the ID can be any identifier, like /user/league or /user/1234.
4. Any other path should produce a 404 “Not found” response.

All four pages should have a common navigation bar that provides links to all pages. (You can just make up what user ID to use – we don’t have an authentication mechanism yet.)

The user profile page should display whatever ID is in the URL, so /user/league could say something like Hello, league but /user/1234 would say Hello, 1234.

Include some content such as an image, list, hyperlinks, etc.

*-- This is some starter code for A11*

```
{-# LANGUAGE OverloadedStrings #-}
```

**module** A11 **where**

```
import qualified Data.ByteString.Char8 as BS
import           Network.HTTP.Types
import           Network.Wai
import qualified Network.Wai.Handler.Warp as Warp
import           Network.Wai.Middleware.RequestLogger (logStdoutDev)
import           Text.Blaze.Html.Renderer.Utf8 (renderHtmlBuilder)
import           Text.Blaze.Html5 as H hiding (main)
import           Text.Blaze.Html5.Attributes as A
```

```
main :: IO ()
main = runOnPort 8080
```

```
runOnPort :: Int -> IO ()
runOnPort port = do
```

```
putStrLn $ "Listening on http://localhost:" <> show port <> "/"
Warp.run port $ logStdoutDev app
```

```
app :: Application
```

```
app request respond = dispatch request >>= respond
```

```
  where
```

```
    dispatch =
```

```
      case pathInfo request of
```

```
        [] -> homePage
```

```
        _ -> return . errorPage notFound404
```

```
buildHtml :: ToMarkup a => Status -> a -> Response
```

```
buildHtml status page =
```

```
  responseBuilder status [(hContentType, "text/html")]
```

```
  (renderHtmlBuilder (toHtml page))
```

```
errorPage :: Status -> Request -> Response
```

```
errorPage status _ =
```

```
  buildHtml status $ h1 $ toHtml $ BS.unpack $ statusMessage status
```

```
homePage :: Request -> IO Response
```

```
homePage _ =
```

```
  return $ buildHtml ok200 homeTemplate
```

```
homeTemplate :: Markup
```

```
homeTemplate = do
```

```
  h1 "My site"
```

```
  p "Hi! This is my portfolio page."
```

```
  p $ do
```

```
    a ! href "/about" $ "About me"
```