

WITS

Mathias Larsen (70434)

Oliver Heinecke (71604)

18. november 2021

1 Introduction

Vi har fået til opgave at lave et hjemmeside der hvor man som bruger skal kunne poste opslag og kunne kommentere andres opslag. Til det har vi et link til vores hjemmeside på: <https://wits.ruc.dk/mamaar/mini-project/login.php> Her har vi designet en hjemmeside som har alle det funktionsræv som er blevet opstillet.

2 Layout

Alle vores dokumenter starter med at kalde *session_start()* og *require_once '/home/mir/lib/db.php'*; da vi skal hente vore API og tillade brug af sessions i alle vores filer.

På vores side har vi en velkomst besked på toppen af siden som enten skriver "welcome [brugers fornavn][brugers efternavn], eller "no user logged in", som er afgjort af om der er en bruger logget ind.

For at danne noget overblik og 'ease of accessability' har vi lavet en navbar i toppen af siden, som indeholder en række knapper. Disse knapper ændres afgjort om en bruger er logget ind. hvis en bruger er logget ind er der en "Home"knapp som sender dig til timeline siden (main.php) og en "create a post"knapp som sender dig til siden hvor du kan lave et post. Under disse knapper er der en drop down menu som hedder profile. Inde i denne dropdown menu er der en "your profile"knapp som sender dig til din side og en log ud

knap. Hvis man ikke er logget ind så ses kun "Home"knappen og en "login" som sender dig til login siden.

Vores hjemmesides hovedstruktur er delt ind som en flexbox som er delt i 3 dele. Den venstre side af siden er 2/12 af side størrelsen og indeholder alle brugerne som er skrevet op til siden. Den midterste del af siden indeholder alle post, og bliver vist på en måde som bliver forklaret i 3. Da dette er hoved delen af siden fylder den 8/12 af siden. Sidst er der en som flex box i højreside af siden som fylder de sidste 2/12 dele af siden.

Vores hjemmeside benytter sig af bootstrap og har derfor et link i bunden af <body> og i toppen af <head>. Dette gør at vi kan benytte os af css fra: <https://getbootstrap.com/docs/5.1/getting-started/introduction/>

3 main.php

I vores timeline viser vi alle vores post fra alle brugere, ligesom andre medie apps som Facebook. Her bruger vi et foreach loop til at vise hver bruger som et link til deres egen profil side ved at bruge en href til hver brugers side. Efter finder vi alle post id og laver et *array_reverse(\$pid)* som vi bruger i et foreach loop igen for at få de sidst tilføjede post først i vores array. Det gør at de nyeste post der bliver lavet er dem der bliver vist først på siden. Det gør vi da det giver mest mening at se de nyeste post i stedet for at skulle helt ned i bunden hver an du vil se et nyt post. Vi gemmer alt specifik data fra det post der skal vises ved brugeres id samt image id. Derefter viser vi den titel på postet ved brug af *\$post_array('title')* og viser navnet på den brugere som poster ved at bruge info fra *spec_bruger.php*. Til hver brugers titel og navn er der selvfølgelig et post som vi viser med *\$post_array['content']*. Hvis man poster et billede med sit post bliver det vist med et if statement som siger at hvis det er et image id køre den foreach statement der giver *get_image(\$i)['path']* som giver billedets path som viser billedet til det tilhørende post.

4 upload.php

Upload.php indeholder en form som modtager en "title", en "content" og "picture". Når submit knappen bliver trykket på vil disse blive gemt i nogle variable, hvor "title" og "content" bliver uploaded ved hjælp af *add_post(\$uid, \$title, \$content)* hvor \$uid er den bruger der uploader posten. Dette kommer fra vores session bruger. \$title, \$content er vores inputs fra formen, som bliver sendt ind igennem en *htmlentities()*, da vi ikke vil ha at bruger kan ødelægge strukturen af programmet ved at skrive html i uploadet.

Hvis brugeren vælger at uploade et billede med sin post, bliver dens "mime type" gemt i \$file_mime_type som så bliver konverteret til et suffix til den path billedet har. Ved aflevering tillader vores side kun 4 filtyper; jpeg, PNG, gif, og jpg. Dette kan udvides. Billedets filepath bliver gemt i variabelen \$filepath og bruges i *add_image(\$filepath, \$filetype)*. For at sørge for at billedet bliver forbundet til den post det blev uploaded med, benytter vi os af *add_attachment(end(get_pids()), end(get_iids()))*. Her er henter vi alle post id'erne for alle postne og alle billederne ved *get_pids()* og *get_iids()*, og vælger det sidste element i arrayet ved *end()*. Vi ved derfor at det sidste billede der blev uploaded og det sidste post der blev uploaded bliver tilknyttet til hinanden.

5 spec_post.php

I spec_post.php har vi en specifik post. Denne side kommer man til hvis man clicker på overskriften af en post og indeholder den post med dens kommentare. Hvis man ikke er logget ind så ser man i toppen af siden en overskriften på posten, med forfatterens navn nedenunder. Forfatterens navn er et link som leder til forfatterens side. Under forfatteren kommer der contenten af posten og endnu under der vises der billeder tilknyttet til posten, hvis der er nogen. Under billederne er der et kommentar felt, som man kan læse hvis man ikke er logget ind, men kun kan tilføje til hvis man er logget ind. Hvis man selv er forfatteren af siden vil der være en "modify post" knap mellem forfatternavnet og titlen som tillader dig at ændre på dit post. Lige så, hvis man er logget ind som forfatteren af posten kan man se en "delete" knap ude

for alle kommentare som gør at man kan slette de specifikke kommentare på sin post. Hvis man er logget ind som en bruger der ikke er forfatteren, så kan man slette sine egne kommentare men ikke andre.

6 spec_bruger.php

I spec bruger viser vi alle post uploaded af denne specifikke bruger. Her kan vi se, ligesom i vores main.php, at alle posts er sorteret efter det nyeste først. Her viser vi alle post som små lister, hvor det første element i listen er overskriften på posten. Siden vi ved at vi er inde på en specifik brugers side, viser vi ikke hvis post det er da vi ved at det er den bruger hvis side vi er inde på. herefter viser vi contenten af posten. vi henter så alle commentare tilknyttet postet og hvis der er nogen viser vi dem under posten ved hjælp af et foreach loop. Hvis der ikke er nogen commentare til postet skrives der "No comments". Disse lister bliver så gentaget for alle post der er blevet uploaded af denne bruger.