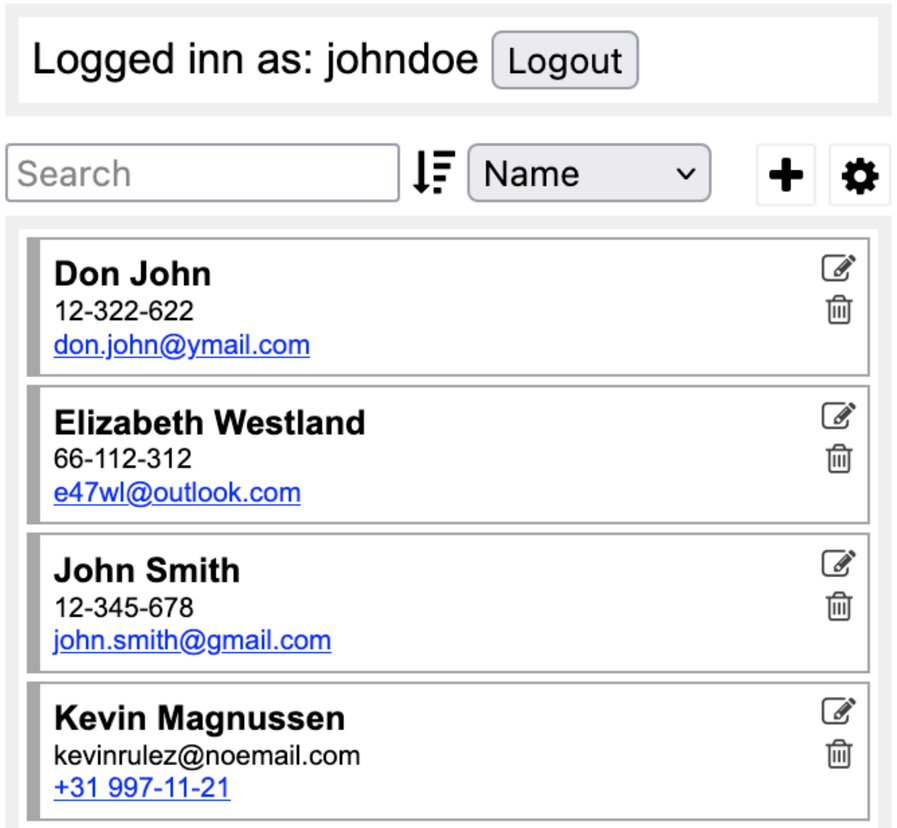
Web Programming
Assignment 8 & Accessability

Leander Jehl | University of Stavanger

Assignment 8 solution walkthrough

assignments/assignment-8/8-solution

- Provide login and backend for contact book



Assignment 8 solution walkthrough

assignments/assignment-8/8-solution

- Database contains
  - Users

```
CREATE TABLE users (
    userid INTEGER,
    username TEXT NOT NULL,
    passwordhash TEXT NOT NULL,
    PRIMARY KEY(userid)
    UNIQUE(username)
)
```

  - Addresses

```
CREATE TABLE addresses (
    addressid INTEGER,
    user INTEGER,
    name TEXT NOT NULL,
    tel TEXT,
    email TEXT,
    PRIMARY KEY(addressid)
    FOREIGN KEY (user) REFERENCES users (userid)
)
```

Assignment 8 solution walkthrough

assignments/assignment-8/8-solution

- **app.py** routes

```
# The first three routes do not fit into a rest API.

@app.route("/")
def index():


@app.route("/login", methods=["POST"])
def login():

@app.route("/logout")
def logout():
```

`/login` returns user data from db.

Assignment 8 solution walkthrough

assignments/assignment-8/8-solution

- **app.py** routes

```
# Here comes the rest API for addresses.
@app.route("/user", methods=["GET"])
def get_user():


@app.route("/addresses", methods=["GET"])
def get_addresses():

@app.route("/addresses", methods=["POST"])
def add_address():

@app.route("/addresses/<addressid>", methods=["PUT"])
def set_address(addressid):

@app.route("/addresses/<addressid>", methods=["DELETE"])
def del_address(addressid):
```

`/user` gets logged in userid from session and returns user data from db.

`/addresses [GET]` returns all addresses for logged in userid

`/addresses [POST]` adds new address and returns new id

`/addresses/<id> [PUT]` updates address

Assignment 8 solution walkthrough

assignments/assignment-8/8-solution

- **data.js** ajax calls for each endpoint

```
async function login(){ ... }

async function logout(){ ... }

// try if there is a user logged in
async function getUser(){... }

async function getAddresses(){... }

async function storeAddress(address){ ... }

async function updateAddress(address){ ... }

async function deleteAddress(address){ ... }
```

Assignment 8 solution walkthrough

assignments/assignment-8/8-solution

- **data.js** ajax calls for each endpoint
- **`catch`** network error
- **`alert`** on error
- **`return true`** or **`false`** to signal success

```
async function deleteAddress(address){
    try {
        let url = "/addresses/" + address.id;
        let response = await fetch(url, {
            method: "DELETE",
        });
        if (response.status == 200){
            return true;
        }
        console.log("Error deleting contact.")
        alert("Error deleting contact, please refresh
page.")
    } catch (error) {
        console.log(error);
        alert(`Network error: ${error}`)
    }
    return false;
}
```

Assignment 8 solution walkthrough

assignments/assignment-8/8-solution

Login flow:

- Check if a user is already logged in:

```
<body onload="getUser()">
```

- Allow for manual log in:

```
<form action="javascript:void(0);" onsubmit="login();">
```

- After login, hide login form - with userid get addresses:

```
async function login(){
   form.style.display = "none";

    try {
        ...
        getAddresses();
   } catch (error) { ... }
}
```

Assignment 8 solution walkthrough

assignments/assignment-8/8-solution

Flaws

- Solution does not show a loading icon, while AJAX requests are ongoing.
- Error messages could be better displayed, not as Alert, but on page.

Accessibility

Accessibility

- Practice of making websites usable to as many people as possible
- People with impaired hearing, sight or motion difficulties, learning disabilities, must be able to use the page

At least 1 in 5 people in the UK have a long term illness, impairment or disability. Many more have a temporary disability.

https://www.gov.uk/guidance/accessibility-requirements-for-public-sector-websites-and-apps

Accessibility

- In Norway, public websites must be universally accessible
- Requirements follow the accessibility standart from the W3C WACG: https://www.w3.org/WAI/standards-guidelines/

Accessibility

- Color blind -> high contrast color schemes
- People with motor difficulties -> Large clicking area
- Limited sight -> Large zoom
- Learning dissabilities - dyslexia -> Use colors and shapes,
  - Support screen readers

Accessibility - Design

- Text must be readable
- Text must be adjustable:
  - Font size
  - Line height
- Contrast
- Usable with large zoom
- …

Accessibility - Keyboard

- Page usable with keyboard only
- No hidden buttons
  - Use button for elements with onclick
- Avoid complex mouse interaction
- Manage focus

```
document.getElementById("mod_name").focus();
```

Accessibility - Screen reader

- Use semantic tags: or use **`role`** attribute
- Use labels
- Use alt for images
- **`area-hidden`**, **`area-label`**, … attributes for dynamic content

Accessibility - Screen reader Example