



FRÓÐSKAPARSETUR  
FØROYA

Vælkomin til  
**Kjakið**

**Tíðindir** Hvat nýtt veitst tú?

**Kjak** Kjak um hvat sum helst.

**Áhugi** Lat heimin vita um tíni áhugamál.

## Verkætlan í Linux Systems

Salomon Vágadal Joensen

Jákup Paulason Olsen

Helena Hentze

## Forum forrit við MariaDB

Ubuntu Server VM/Ansible stýring

Vegleiðari: Jóhannus Kristmundsson

Innlatin: 1. marts 2024



NÁTTÚRUVÍSINDAEILDIN

Heiti / Title ***Forum forrit við MariaDB /  
Forum software with MariaDB***

Høvundar / Authors Salomon Vágadal Joensen  
Jákup Paulason Olsen  
Helena Hentze

Vegleiðari / Supervisor Jóhannus Kristmundsson

Ritslag / Report Type Uppgáva í Linux Systems  
Latið inn / Submitted 1. mars 2020

NVDRit

© Náttúruvísindadeildin og høvundarnir 2024  
ISSN 1601-9741

Útgevri / Publisher Náttúruvísindadeildin, Fróðskaparsetur Føroya  
Bústaður / Address Nóatún 3, FO 100 Tórshavn, Føroyar (Faroe Islands)  
Tlf, Fax, Teldupostur +298 352 550 • +298 352 551 • nvd@setur.fo

## Contents

<b>1</b>	<b>Samandrátt (Abstract)</b>	<b>4</b>
<b>2</b>	<b>Trupulleika-orðing</b>	<b>4</b>
<b>3</b>	<b>Mál</b>	<b>4</b>
<b>4</b>	<b>Framgangsháttur</b>	<b>4</b>
<b>5</b>	<b>Design</b>	<b>5</b>
<b>6</b>	<b>Lýsing av Loysn</b>	<b>5</b>
6.1	Web Uppseting . . . . .	5
6.2	Databasa Uppseting . . . . .	7
6.3	Heimasíðan . . . . .	7
6.4	Ansible . . . . .	8
<b>7</b>	<b>Perspektivisering</b>	<b>9</b>
<b>8</b>	<b>Niðurstøða</b>	<b>9</b>
<b>9</b>	<b>Appendix</b>	<b>9</b>
9.1	Tíðarætlan . . . . .	9
	kjakdb Schema . . . . .	10
	local.yml . . . . .	16
	update_kjakdb.sh.j2 . . . . .	21
	conn.php.j2 . . . . .	26
	create_thread.php . . . . .	27
	index.php . . . . .	30
	script.js . . . . .	32
	style.css . . . . .	36
	view_forum.php . . . . .	43
	view_thread.php . . . . .	48

## 1 Samandrátt (Abstract)

Í hesum rapporti stendur hvussu vit gjørdur eina kjak heimasíðu, hvat fyri tól-menni (harímillum Ansible, Apache2, MariaDB og PHP) vit brúktu og hvussu arkitektururin var uppbygður til at fremja hetta í verki. Eisini hugdi vit at hvussu front-síðan av heimasíðuni verður tillagur, so at tað er lætt atkomiligt og gevur brúkarinum yvirlit av heimasíðuni.

## 2 Trupulleika-orðing

Hvussu ger man eina kjak heimasíðu sum fólk kunnu vitja og stovna tráðir og leggja innlegg í? Og møguliga eisini hava møguleika at deila media har?

- Man má gera sær greitt at har má vera ein heimasíða, sum fólk kunnu vitja.
- Har má vera ein *backend* har ein dátagrunnur er.
- Har má vera eitt *millumlið* sum tulkar millum *backend* og *frontend* (t.v.s. heimasíðan).
- Har má vera ein *frontend* sum er brúkaraflatan fyri kjak heimasíðuna.

## 3 Mál

- Tað má vera lætt at vita hvar man er og hvussu man kemur fram til brúkarin vil vera.
- Á hesari heimasíðuni skal brúkarin kunna síggja kjaksíður.
  - Fáa yvirlit av kjak undirsíðunum.
  - Fara inn á eina kjak undirsíðu.
  - Síggja tráðir og kunna stovna tráðir.
  - Kunna fara inn á einkultar tráðir og svara í einum tráði og viðmerkja navn, tekstsvor og um tey vilja leggja mynd avtrat.

## 4 Framgangsháttur

Vit byrja við einari stutta analysu hvussu hetta skal fremjast.

- Arkitektur bygnaða av probleminum og hvussu tað fer at síggja út.
- Gera ein databasa í MariaDB har man kann stovna ein *thread* í 3 ymiskum kjakforum har fólk kunnu svara uppá.
- Við einum fullfiggjaðum MariaDB datagrunn, byggja eina heimasíðu sum virkar sum eitt *interface* millum heimasíðuna og datagrunnin.

- *Business logic* millumlið verður brúkt PHP til samskipting millum heimasíðuna og MariaDB.
- Millumliðið fer at avgera hvussu úrslit frá datagrunninum verður víst.
- HTML verður gjørt fyri at fáa grund bygnaðin av síðuni.
- CSS verður nýtt til at pynta HTML og gera snið
- JS leggur síðani funktionlatitet til sum ikki hevur við dátabasan at gera
- PHP byggur dynamiskt síðuna við at taka postar, tráðir, svør úr dátabasanum og leggur teir í HTML á síðuni.

## 5 Design



Figure 1: Samskipti millum heimasíðu og datagrunnin.

Ein Ubuntu Server við einari lokalari heimasíðu og brúkt ein MariaDB dátagrúnn at goyma tráðirnir og postar í. PhpMyAdmin verður brúkt til at síggja dátagrúnnin og tað er installera á sjálva serveran, men er ikki partur av Ansible playbook, tí tað er ikki neyðugt fyri at fáa kjak heimasíðuna at virka.

HTML verður brúkt til at leggja grund til útsjónina, og so at PHP og JS kunnu peika til ávis støð tá heimasíðan verður bygt. CSS verður nýtt til at pynta á HTML, fyri at fáa eina pena og lesiliga heimasíðu. JS verður nýtt fyri at fáa funktionalitet minka og forstørta myndir, og vísa/fjala svar formular. PHP verður brúkt til at dynamiskt byggja lutir av heimasíðuni, sum t.d. postar, svør og tráðir.

## 6 Lýsing av Loysn

### 6.1 Web Uppseting

Lendingar síðan gevur yvirlit av teimum kjakforum sum eru, og leinkir til tey. Brúkarin velur tað forum teimum ynskir, og verður síðani koyrdur har til. Á kjakforum kunnu tey hyggja gjøgnum postar, síggja svør og fara ígjøgnum tráðir, samt sum at leggja egnir postir út ella leggja svar til aðrar postir og onnur svør. Fyri at síggja øll svør má brúkarin fara inn á postin, og um hann ynskir at síggja tráðir undir svørum til postin trýstir hann á ein tráð fyri at síggja tann tráðin, og mesta lagið vera víst 10 tráðir per forum og 10 svør per tráð. Litir eru valdir

fyri at geva brúkarinum eina róliga kenslu, og grønt er ofta sett í samband við ró. Stórar yvirskriftir og stórir knappar eru valdir fyri at vera lættir at lesa.

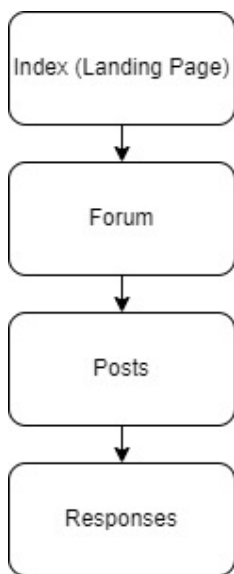


Figure 2: Hvussu uppsetingin av forum er.

## 6.2 Databasa Uppseting

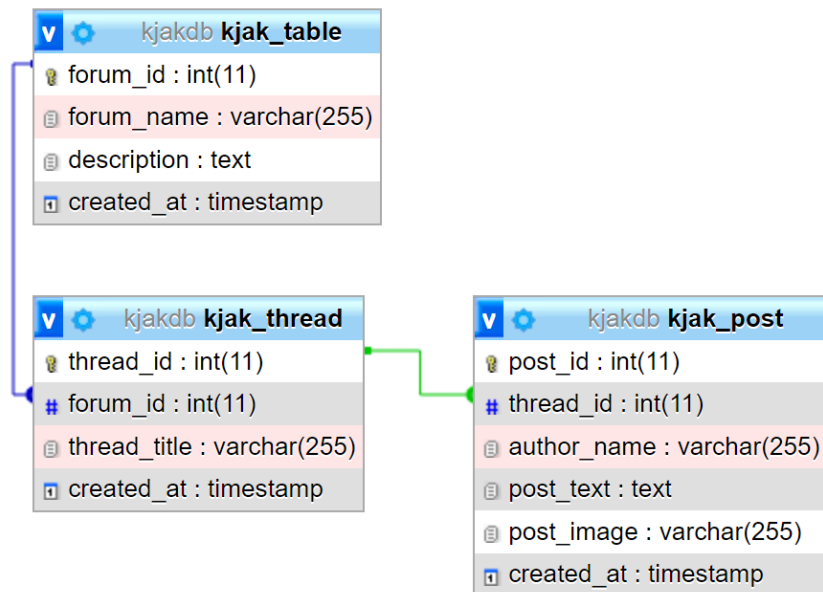


Figure 3: Databasa relatióinirnar fyri *kjakdb*

Databasa relatióinirnar vísa hvussu tær 3 tabelirnar eru relatarar. **kjak\_table** er høvuðs tabellin og hevur *Primary Key* `forum_id` til *Foreign Key* til **kjak\_thread**, og **kjak\_thread** hevur *Primary Key* `thread_id` til *Foreign Key* til **kjak\_post**.

## 6.3 Heimasíðan

Heimasíðan er bert 4 síður við embedaða PHP kodu til at vísa dáta frá *kjakdb* dátagrunnins 3 tabellir. Annars er ein CSS fil *style.css* fyri uppseting og ein *conn.php* við íbinding upplýsingarnar til dátagrunnin. Eisini er ein JS fil *script.js* fyri interaktivar funktiónir. Tær 4 síðurnar eru:

- *index.php* Heimasíðan sum vísur forumini.
- *view\_forum.php* Undirsíðan fyri at vísa tráðir fyri eitt forum.
- *view\_thread.php* Undirsíðan fyri at vísa ein á tráð og allar svør í tí tráðinum.
- *create\_thread.php* Fyri at stovna ein nýggjan tráð.

## 6.4 Ansible

Hesar 7 fílar (og eisini 8nda *favicon.ico* filin) verða stovnaðir av at koyra kommandoina á einari VM við *Ansible*:

```
ansible-pull -U https://github.com/salomonvjoensen/linuxskipanir.git
```

Tað ekskeverarar eina *Ansible* Playbook **local.yml** á hasum repository sum ger her hesu trin idempotentli (t.v.s. kann vera endurtikið uttan at bróta uppá nakað ella gera óneyðug kopiir):

- Promptar brúkaran fyri -
  - MySQL brúkaranavn.
  - MySQL kodu.
  - kjakdb brúkaranavn.
  - kjakdb kodu.
- Installerar Apache2 og relevant PHP og MySQL modulir, startar Apache2.
- Ger eina *uploads* mappu í */var/www/html/uploads*
- Umdoypir Apache2 standard *index.html* filuna (um hon er til) til *index.html.old*.
- Tær 7 fílar kopieraðar yvir til *html* mappuna.
- *conn.php.j2* template filin við kjakdb brúkaranavn og kodu verður framleitt og stovna.
- **MariaDB** tænasta verður stovna og byrja.
- **Pip** og **PyMySQL** verða installera.
- Stovna *MySQL brúkaran* og *root brúkaran* (við sama MySQL kodu)
- Stovna *kjakdb* dátagrunnin.
- Stovna *kjakdb brúkaran* fyri dátagrunnin.
- Koyra eitt *bash* script til at dagføra dátagrunnin við teim 3 tabellir og 3 forum INSERT.
- Seinast seta eitt *setup flag* til, so man verður ikki promptaður eina aðru ferð man koyrir hesa **Ansible playbook**.

Aftaná kann man opna localhost í ein kagara og Kjak heimasíðan er uppi og koyrir.



## 7 Perspektivisering

Heimasíðan hefur tað mest einklu treytirnar fyri at vera hugsa sum ein kjak-heimasíða, Sígga nøkur forum, tráðir, stovna tráðir, svara á tráðum og leggja myndir út saman við svør.

Tað er ein einkul kjak heimasíða, sum kundi verið nógv útbygt, mest sannlíkandi kundi verið at vitjandi kundu stovna brúkarar (um tey vildu), so tey sjálvu kundu strika teirra egnu tráðir og svør. Ting sum at indeksa heimasíðuna, so man kundi leita uppá dátagrunnin kundi eisini verið implementera.

Tað er skjótt at fáa heimasíðuna upp at koyra bara við einari einklari bash kommandoina, og 4 promptir har man kann seta sjálv(ur) hvat fyri brúkaranøvn og kodir man vil brúka, givið man hefur **Ansible** á einum Linux líknandi umhvørvi (vit brúktu Ubuntu Server VM), og tað vísur eisini styrkina í **Ansible**.

Vit hugsaðu um at brúka **Ansible Playbook**, so man kundi havt forriti koyrt á fleiri umhvørvum men hvat er meiningin at hava fraktuera kjak á nógvum støðum, tá man kann hava eitt samla stað at kjakast í?

## 8 Niðurstøða

Okkara arbeiði og skeið gav okkum innlit hvussu man brúkar Linux umhvørvi og hvussu tað er at arbeiða næstan bara í einum terminal uppseting. Nógvar kommandoir skuldi man læra, og brýtir nógv frá tí vanda GUI umhvørvi man kennur mest frá Windows, men um man dugir kann man automatisera øgiliga nógv og næstan hálv-forrita redigering av fílum, her hugsi eg um *vim* editorin.

Kjak heimasíðan er einkul, men tøkkin aftanfyrir er sørmi ikki. Man kundi sett upp heimasíðuna uttan at hava **Ansible**, men tað hevði kravt nógv manuelt arbeiði hvørja ferð man skuldi sett upp eitt nýtt kjak heimasíðu umhvørvi aðra staðnis.

## 9 Appendix

### 9.1 Tíðarætlan

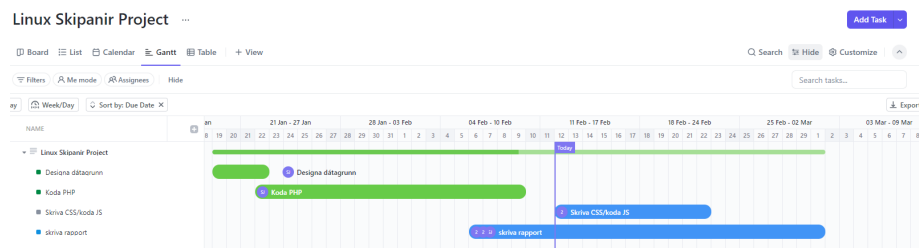


Figure 4: Skermmynd tikin av tíðarætlan mánadagin 12. februar 2024

Fyrst var dátagrunnurinn designaður, so bleiv PHP kodan koda til dátagrunnin. Meðan tað varð arbeiða uppá tað byrjaðu vit so smátt at skriva rapportina og gera tað seinasta hondverki av HTML, CSS & JS uppseting av heimasíðuna, bara so hon sær eitt sindur vøkur út.

---

**PDF-eksportside**

Indholdsfortegnelse
---------------------

1 kjak_post	Side nummer: 2
2 kjak_table	Side nummer: 3
3 kjak_thread	Side nummer: 4
4 Relationel skematik	Side nummer: 5

PDF-eksportside

1 kjak\_post

Oprettelse: 07. 02 2024 kl. 21:40:43  
Seneste opdatering: 09. 02 2024 kl. 12:49:27

Kolonne	Datatype	Attributter	Nulværdi	Standardværdi	Ekstra	Linker til	Kommentarer	MIME
post_id	int(11)		Nej		auto_increment			
thread_id	int(11)		Ja	NULL		-> kjak_thread.thread_id ON UPDATE RESTRICT ON DELETE CASCADE		
author_name	varchar(255)		Ja	NULL				
post_text	text		Nej					
post_image	varchar(255)		Ja	NULL				
created_at	timestamp		Nej	current_timestamp()				

PDF-eksportside

2 kjak\_table

Oprettelse: 07. 02 2024 kl. 21:40:43

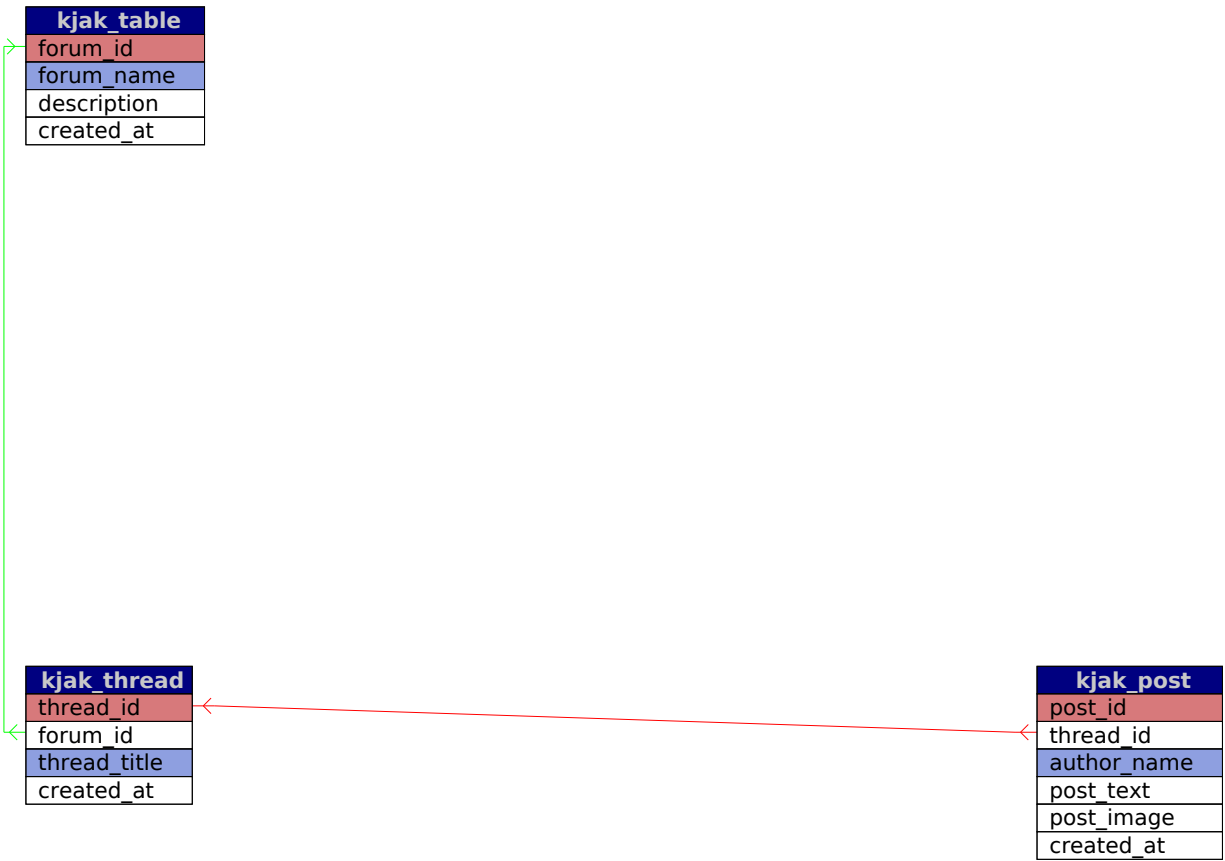
Kolonne	Datatype	Attributter	Nulværdi	Standardværdi	Ekstra	Linker til	Kommentarer	MIME
forum_id	int(11)		Nej		auto_increment			
forum_name	varchar(255)		Nej					
description	text		Ja	NULL				
created_at	timestamp		Nej	current_timestamp()				

PDF-eksportside

3 kjak\_thread

Oprettelse: 07. 02 2024 kl. 21:40:43  
Seneste opdatering: 09. 02 2024 kl. 12:42:52

Kolonne	Datatype	Attributter	Nulværdi	Standardværdi	Ekstra	Linker til	Kommentarer	MIME
thread_id	int(11)		Nej		auto_increment			
forum_id	int(11)		Ja	NULL		-> kjak_table.forum_id ON UPDATE RESTRICT ON DELETE CASCADE		
thread_title	varchar(255)		Nej					
created_at	timestamp		Nej	current_timestamp()				



# Local.yml

```
---
- hosts: localhost
  become: yes

  tasks:
    - name: Check for setup complete flag file
      ansible.builtin.stat:
        path: /var/www/html/.setup_complete
      register: setup_flag # Store the result

    - name: Initialize setup_complete flag (true if missing)
      ansible.builtin.set_fact:
        setup_complete: "{{ setup_flag.stat.exists }}"

    - name: Gather credentials (if setup incomplete)
      block:
        - pause:
            prompt: "Enter your desired MySQL username"
            register: mysql_credentials
        - set_fact:
            mysql_user: "{{ mysql_credentials.user_input }}"

        - pause:
            prompt: "Enter your desired MySQL password"
            register: mysql_password
        - set_fact:
            mysql_password: "{{ mysql_password.user_input }}"

        - pause:
            prompt: "Enter your desired kjakdb username"
            register: anon_user_credentials
        - set_fact:
            anon_user: "{{ anon_user_credentials.user_input }}"

        - pause:
            prompt: "Enter your desired kjakdb user password"
            register: anon_password
        - set_fact:
            anon_password: "{{ anon_password.user_input }}"
      when: not setup_complete

    - name: Install Apache
      ansible.builtin.apt:
        name: apache2
        state: present
```



# Local.yml

```
- name: Install PHP and Apache module
  ansible.builtin.apt:
    name:
      - php
      - libapache2-mod-php
    state: present

- name: Install PHP MySQL extension
  ansible.builtin.apt:
    name: php-mysql
    state: present

- name: Restart Apache (to load PHP module)
  ansible.builtin.service:
    name: apache2
    state: restarted

- name: Create uploads folder
  ansible.builtin.file:
    path: /var/www/html/uploads
    state: directory
    owner: root
    group: root
    mode: '0777'

- name: Check if index.html exists
  ansible.builtin.stat:
    path: /var/www/html/index.html
  register: index_file_stats

- name: Rename index.html to index.html.old (if it exists)
  ansible.builtin.file:
    src: /var/www/html/index.html
    dest: /var/www/html/index.html.old
    state: file
  when: index_file_stats.stat.exists

- name: Copy web files
  ansible.builtin.copy:
    src: "www/{{ item }}"
    dest: /var/www/html/{{ item }}
  with_items:
    - index.php
    - favicon.ico
```

# Local.yml

```
- style.css
- script.js
- create_thread.php
- view_forum.php
- view_thread.php

- name: Create conn.php from template
  ansible.builtin.template:
    src: www/conn.php.j2
    dest: /var/www/html/conn.php
    when: not setup_complete

- name: Install MariaDB server
  ansible.builtin.apt:
    name: mariadb-server
    state: present

- name: Start and enable MariaDB service
  ansible.builtin.service:
    name: mariadb
    state: started
    enabled: true

- name: Ensure pip is installed
  ansible.builtin.apt:
    name: python3-pip
    state: present

- name: Install PyMySQL
  ansible.builtin.pip:
    name: PyMySQL
    state: present

- name: Set MySQL root password
  ansible.builtin.shell: |
    mysql -u root -e "SET PASSWORD FOR root@'localhost' = PASSWORD('{{
mysql_password }}');"
  args:
    executable: /bin/bash
    when: not setup_complete

- name: Set password for MySQL user (including root)
  ansible.builtin.mysql_user:
    login_user: root
    login_password: "{{ mysql_password }}"
```

# Local.yml

```
    check_implicit_admin: true
#    login_unix_socket: /var/lib/mysql/mysql.sock
    name: "{{ mysql_user }}"
    password: "{{ mysql_password }}"
    priv: " *.*:ALL"
    host: localhost
when: not setup_complete

- name: Create .my.cnf for MySQL authentication
  ansible.builtin.copy:
    dest: "~/.my.cnf"
    content: |
      [client]
      user={{ mysql_user }}
      password={{ mysql_password }}
    mode: '0600'
  when: not setup_complete

- name: Ensure the database 'kjakdb' exists
  ansible.builtin.mysql_db:
    name: kjakdb
    state: present
    login_user: "{{ mysql_user }}"
    login_password: "{{ mysql_password }}"
  when: not setup_complete

- name: Ensure the kjakdb user exists with privileges
  ansible.builtin.mysql_user:
    login_user: root
    login_password: "{{ mysql_password }}"
    name: "{{ mysql_user }}"
    password: "{{ mysql_password }}"
    priv: 'kjakdb.*:ALL'
    host: localhost
    state: present
  when: not setup_complete

- name: Create update script from template
  ansible.builtin.template:
    src: sh/update_kjakdb.sh.j2
    dest: /tmp/update_kjakdb.sh
    mode: '0755'
  when: not setup_complete

- name: Execute the update script
```

## Local.yml

```
ansible.builtin.command:
  cmd: /tmp/update_kjakdb.sh
when: not setup_complete

- name: Create state file after initial setup
  ansible.builtin.file:
    path: /var/www/html/.setup_complete
    state: touch
    mode: '0644'
  when: not setup_complete

- name: Set initial setup flag
  ansible.builtin.set_fact:
    setup_complete: true
```

# Update\_kjakdb.sh.j2

```
#!/bin/bash

DATABASE='kjakdb'
MYSQL_ROOT_USER=root
MYSQL_ROOT_PASSWORD='{{ mysql_password }}'
ANON_USER='{{ anon_user }}'
ANON_PASSWORD='{{ anon_password }}'

# Function to check if user exists and create if not
ensure_user_exists() {
    local user=$1
    local password=$2

    echo "Checking if user $user exists..."
    if ! mysql -u "$MYSQL_ROOT_USER" -p"$MYSQL_ROOT_PASSWORD" -e "SELECT 1 FROM
mysql.user WHERE user = '$user'" | grep -q 1; then
        echo "User $user does not exist, creating..."
        mysql -u "$MYSQL_ROOT_USER" -p"$MYSQL_ROOT_PASSWORD" -e "CREATE USER
'$user'@'localhost' IDENTIFIED BY '$password';"
    else
        echo "User $user already exists."
    fi
}

# Function to create a single table
create_table() {
    local table_name=$1
    local columns=$2 # Example: "forum_id int(11) NOT NULL AUTO_INCREMENT,
forum_name varchar(255) NOT NULL, description text DEFAULT NULL, created_at
timestamp NOT NULL DEFAULT current_timestamp()"
    local primary_key=$3

    echo "Checking if table $table_name exists..."
    if ! mysql -u "$MYSQL_ROOT_USER" -p"$MYSQL_ROOT_PASSWORD" -D "$DATABASE" -
sse "SHOW TABLES LIKE '$table_name';" | grep -q "$table_name"; then
        echo "Creating table $table_name..."
        mysql -u "$MYSQL_ROOT_USER" -p"$MYSQL_ROOT_PASSWORD" -D "$DATABASE" -e
"CREATE TABLE $table_name ($columns, PRIMARY KEY ($primary_key));"
    else
        echo "Table $table_name already exists."
    fi
}

# Grant permissions to user
grant_permissions() {
    local user=$1
```

# Update\_kjakdb.sh.j2

```
local database=$2
local table=$3
local privileges=$4

echo "Granting $privileges privileges to $user on $database..."
mysql -u "$MYSQL_ROOT_USER" -p"$MYSQL_ROOT_PASSWORD" -e "GRANT $privileges
ON $database.$table TO '$user'@'localhost';"
mysql -u "$MYSQL_ROOT_USER" -p"$MYSQL_ROOT_PASSWORD" -e "FLUSH PRIVILEGES;"
}

# Function to check and add primary keys and indexes
add_primary_key_and_index() {
    local table=$1
    local primary_key=$2
    local index_columns=$3

    echo "Checking primary key for table $table..."
    if ! mysql -u "$MYSQL_ROOT_USER" -p"$MYSQL_ROOT_PASSWORD" -D "$DATABASE" -
sse "SHOW KEYS FROM $table WHERE Key_name = 'PRIMARY';" | grep -q 'PRIMARY';
then
        echo "Adding primary key ($primary_key) to $table..."
        mysql -u "$MYSQL_ROOT_USER" -p"$MYSQL_ROOT_PASSWORD" -D "$DATABASE" -e
"ALTER TABLE $table ADD PRIMARY KEY ($primary_key);" || {
            # Handle duplicate key error (MySQL error code 1062)
            if [[ $? -eq 1062 ]]; then
                echo "Error adding primary key to $table: Duplicate Entry '0' for
key 'PRIMARY'"
            else
                # Re-raise the error if it's something else
                echo "Error setting primary key in $table: $?"
                exit 1
            fi
        }
    else
        echo "Primary key ($primary_key) already exists in $table."
    fi

    if [[ -n "$index_columns" ]]; then
        for index_column in ${index_columns//,/ }; do
            echo "Checking index for column $index_column in $table..."
            if ! mysql -u "$MYSQL_ROOT_USER" -p"$MYSQL_ROOT_PASSWORD" -D "$DATABASE"
-sse "SHOW INDEX FROM $table WHERE Column_name = '$index_column';" | grep -q
"$index_column"; then
                echo "Adding index for column $index_column in $table..."
                mysql -u "$MYSQL_ROOT_USER" -p"$MYSQL_ROOT_PASSWORD" -D "$DATABASE" -e
"ALTER TABLE $table ADD INDEX ($index_column);"
```

# Update\_kjakdb.sh.j2

```
        else
            echo "Index for column $index_column already exists in $table."
        fi
    done
fi
}

# Function to check and add foreign keys
add_foreign_key() {
    local table=$1
    local constraint_name=$2
    local foreign_key=$3
    local references=$4

    echo "Checking foreign key $constraint_name in table $table..."
    if ! mysql -u "$MYSQL_ROOT_USER" -p"$MYSQL_ROOT_PASSWORD" -D "$DATABASE" -s -e "SELECT CONSTRAINT_NAME FROM information_schema.TABLE_CONSTRAINTS WHERE CONSTRAINT_SCHEMA = '$DATABASE' AND TABLE_NAME = '$table' AND CONSTRAINT_NAME = '$constraint_name';" | grep -q "$constraint_name"; then
        echo "Adding foreign key $constraint_name to $table..."
        mysql -u "$MYSQL_ROOT_USER" -p"$MYSQL_ROOT_PASSWORD" -D "$DATABASE" -e "ALTER TABLE $table ADD CONSTRAINT $constraint_name FOREIGN KEY ($foreign_key) REFERENCES $references;"
    else
        echo "Foreign key $constraint_name already exists in $table."
    fi
}

# Function to check and set AUTO_INCREMENT
set_auto_increment() {
    local table=$1
    local column=$2

    # Check if AUTO_INCREMENT is already set
    if ! mysql -u "$MYSQL_ROOT_USER" -p"$MYSQL_ROOT_PASSWORD" -D "$DATABASE" -s -e "SELECT AUTO_INCREMENT FROM TABLES WHERE TABLE_SCHEMA = '$DATABASE' AND TABLE_NAME = '$table' AND COLUMN_NAME = '$column';" | grep -q "$column"; then
        echo "Setting AUTO_INCREMENT for $column in $table..."
        mysql -u "$MYSQL_ROOT_USER" -p"$MYSQL_ROOT_PASSWORD" -D "$DATABASE" -e "ALTER TABLE $table MODIFY $column int(11) NOT NULL AUTO_INCREMENT;"
    else
        echo "$column in $table is already set to AUTO_INCREMENT."
    fi
}

# Ensure 'anon' user exists and grant permissions
```

# Update\_kjakdb.sh.j2

```
ensure_user_exists "$ANON_USER" "$ANON_PASSWORD"
grant_permissions "$ANON_USER" "$DATABASE" "*" "SELECT"
grant_permissions "$ANON_USER" "$DATABASE" "kjak_thread" "INSERT, UPDATE"
grant_permissions "$ANON_USER" "$DATABASE" "kjak_post" "INSERT, UPDATE"

# Create the tables idempotently
create_table 'kjak_table' 'forum_id int(11) NOT NULL AUTO_INCREMENT,
forum_name varchar(255) NOT NULL, description text DEFAULT NULL, created_at
timestamp NOT NULL DEFAULT current_timestamp()' 'forum_id'
create_table 'kjak_thread' 'thread_id int(11) NOT NULL AUTO_INCREMENT,
forum_id int(11) DEFAULT NULL, thread_title varchar(255) NOT NULL, created_at
timestamp NOT NULL DEFAULT current_timestamp()' 'thread_id'
create_table 'kjak_post' 'post_id int(11) NOT NULL AUTO_INCREMENT, thread_id
int(11) DEFAULT NULL, author_name varchar(255) DEFAULT NULL, post_text text
NOT NULL, post_image varchar(255) DEFAULT NULL, created_at timestamp NOT NULL
DEFAULT current_timestamp()' 'post_id'

# Add primary keys, indexes, and set auto-increment
add_primary_key_and_index 'kjak_post' 'post_id' 'thread_id'
add_primary_key_and_index 'kjak_table' 'forum_id'
add_primary_key_and_index 'kjak_thread' 'thread_id' 'forum_id'

set_auto_increment 'kjak_post' 'post_id'
set_auto_increment 'kjak_table' 'forum_id'
set_auto_increment 'kjak_thread' 'thread_id'

# Add foreign keys
add_foreign_key 'kjak_post' 'kjak_post_ibfk_1' 'thread_id'
'kjak_thread(thread_id)'
add_foreign_key 'kjak_thread' 'kjak_thread_ibfk_1' 'forum_id'
'kjak_table(forum_id)'

# Insert the forum data if it doesn't exist
mysql -u "$MYSQL_ROOT_USER" -p"$MYSQL_ROOT_PASSWORD" -D "$DATABASE" -e "INSERT
INTO kjak_table (forum_name, description)
SELECT * FROM (SELECT 'Tíðindir', 'Hvat nýtt veitst tú?') AS tmp
WHERE NOT EXISTS (
  SELECT forum_name FROM kjak_table WHERE forum_name = 'Tíðindir'
) LIMIT 1;

INSERT INTO kjak_table (forum_name, description)
SELECT * FROM (SELECT 'Kjak', 'Kjak um hvat sum helst.') AS tmp
WHERE NOT EXISTS (
  SELECT forum_name FROM kjak_table WHERE forum_name = 'Kjak'
) LIMIT 1;
```



## Update\_kjakdb.sh.j2

```
INSERT INTO kjak_table (forum_name, description)
SELECT * FROM (SELECT 'Áhugi', 'Lat heimin vita um tíni áhugamál.') AS tmp
WHERE NOT EXISTS (
    SELECT forum_name FROM kjak_table WHERE forum_name = 'Áhugi'
) LIMIT 1;
"
```

## conn.php.j2

```
<?php
$host = '127.0.0.1';
$db = 'kjakdb';
$user = '{{ anon_user }}';
$password = '{{ anon_password }}';
$charset = 'utf8mb4';

$dsn = "mysql:host=$host;dbname=$db;charset=$charset";
$options = [
    PDO::ATTR_ERRMODE            => PDO::ERRMODE_EXCEPTION,
    PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
    PDO::ATTR_EMULATE_PREPARES   => false,
];

try {
    $pdo = new PDO($dsn, $user, $password, $options);
} catch (\PDOException $e) {
    throw new \PDOException($e->getMessage(), (int)$e->getCode());
}
?>
```

# Create\_thread.php

```
<?php
include 'conn.php'; // Make sure this points to your actual database
connection script

$forum_name = isset($_GET['forum_name']) ? urldecode($_GET['forum_name']) :
''; // Initialize forum_name variable

// Check if the form has been submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $pdo->beginTransaction(); // Start transaction

    try {
        // Extract and sanitize input
        $forum_id = isset($_POST['forum_id']) ? (int)$_POST['forum_id'] : 0;
        $thread_title = filter_input(INPUT_POST, 'thread_title',
FILTER_SANITIZE_STRING);
        $author_name = filter_input(INPUT_POST, 'author_name',
FILTER_SANITIZE_STRING) ?? 'Anonymous'; // Default to 'Anonymous' if empty
        $post_text = filter_input(INPUT_POST, 'post_text',
FILTER_SANITIZE_STRING);
        $created_at = date('Y-m-d H:i:s');

        // Insert new thread
        $sqlThread = "INSERT INTO kjak_thread (forum_id, thread_title,
created_at) VALUES (:forum_id, :thread_title, :created_at)";
        $stmtThread = $pdo->prepare($sqlThread);
        $stmtThread->execute(['forum_id' => $forum_id, 'thread_title' =>
$thread_title, 'created_at' => $created_at]);
        $thread_id = $pdo->lastInsertId();

        // Insert opening post without image first
        $sqlPost = "INSERT INTO kjak_post (thread_id, author_name, post_text,
created_at) VALUES (:thread_id, :author_name, :post_text, :created_at)";
        $stmtPost = $pdo->prepare($sqlPost);
        $stmtPost->execute(['thread_id' => $thread_id, 'author_name' =>
$author_name, 'post_text' => $post_text, 'created_at' => $created_at]);
        $post_id = $pdo->lastInsertId();

        // Handle file upload
        if (!empty($_FILES["post_image"]["name"])) {
            $target_dir = "uploads/";
            $fileExtension = pathinfo($_FILES["post_image"]["name"],
PATHINFO_EXTENSION);
            // Extract the original filename without the extension
```

# Create\_thread.php

```
        $originalFilename = pathinfo($_FILES["post_image"]["name"],
PATHINFO_FILENAME);
        // Sanitize the filename to remove spaces and special characters
        $safeFilename = preg_replace("/[^\a-zA-Z0-9_-]/", "",
$originalFilename);
        // Construct the new filename including the directory, the
sanitized original filename, the post ID, and the file extension
        $newFileName = $safeFilename . "_" . $post_id . "." .
$fileExtension;
        $target_file = $target_dir . $newFileName;

        if (move_uploaded_file($_FILES["post_image"]["tmp_name"],
$target_file)) {
            // Successfully uploaded, update post record with new image
path
            $sqlUpdatePost = "UPDATE kjak_post SET post_image =
:post_image WHERE post_id = :post_id";
            $stmtUpdatePost = $pdo->prepare($sqlUpdatePost);
            $stmtUpdatePost->execute(['post_image' => $target_file,
'post_id' => $post_id]);
        } else {
            throw new Exception("Failed to upload file.");
        }
    }

    $pdo->commit(); // Commit transaction
    header("Location: view_thread.php?thread_id=" . $thread_id .
"&forum_id=" . $forum_id); // Redirect to the newly created thread
    exit();
} catch (Exception $e) {
    $pdo->rollback(); // Rollback transaction on error
    echo "Error: " . $e->getMessage();
}
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Stovna tr       <?php echo htmlspecialchars($forum_name); ?></title>
    <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
    <div class="nav-bar">
```

# Create\_thread.php

```
<!-- "Go Back" Button -->
<button onclick="goBackStructured()">Go Back</button>
</div>
<div id="forum" class="forum">
  <div class="small-title">Stovna nýggjan </div>
  <div class="title"><?php echo htmlspecialchars($forum_name); ?></div>
</div>
<br>
<form action="create_thread.php" method="post" enctype="multipart/form-
data">
  <input type="hidden" name="forum_id" value="<?php echo
$_GET['forum_id']; ?>">
  <label for="thread_title">Tráð Navn:</label>
  <input type="text" id="thread_title" name="thread_title"
required><br><br>
  <label for="author_name">Títt Navn (valfrítt):</label>
  <input type="text" id="author_name" name="author_name"><br><br>
  <label for="post_text">Tekstur: </label>
  <textarea id="post_text" name="post_text" required></textarea><br><br>
  <label for="post_image">Viðheft mynd (valfrítt):</label>
  <input type="file" id="post_image" name="post_image"><br><br>
  <input type="submit" value="Create Thread">
</form>
<script src="script.js"></script>
</body>
</html>
```

# index.php

```
<?php
include 'conn.php'; // Include your database connection file

try {
    $sql = "SELECT forum_id, forum_name, description FROM kjak_table";
    $stmt = $pdo->prepare($sql);
    $stmt->execute();
    $forums = $stmt->fetchAll();
} catch (Exception $e) {
    die("Error: " . $e->getMessage());
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Kjakið</title>
    <link rel="stylesheet" type="text/css" href="style.css">
</head>

<div class="nav-bar">

</div>

<body>
    <div id="forum" class="forum">
        <div class="small-title">Vælkomin til</div>
        <div class="title">Kjakið</div>
    </div>
    <div id="forums-container">
        <?php foreach ($forums as $forum): ?>
            <div class="forum">
                <a class="forum-link" href="view_forum.php?forum_id=<?php echo
htmlspecialchars($forum['forum_id']); ?>&forum_name=<?php echo
urlencode($forum['forum_name']); ?>">
                    <?php echo htmlspecialchars($forum['forum_name']); ?>
                </a>
                <?php echo htmlspecialchars($forum['description']); ?>
            </div>
        <?php endforeach; ?>
    </div>
</body>
</html>
```

index.php

# script.js

```
document.addEventListener('DOMContentLoaded', function() {
    // Initialize post loading and setup. If using static content, this can be
    // omitted or adjusted.
    // fetchPosts(); // Uncomment if dynamically loading posts.
    adjustResponsesVisibility(); // Ensure only the first five responses are
    // visible on load.
    setupImageToggle(); // Setup click-to-toggle size functionality for
    // images.
});

// Function to fetch and display posts dynamically. Adjust as needed for your
// application.
function fetchPosts() {
    fetch('fetch_posts.php')
        .then(response => response.json())
        .then(posts => {
            const forum = document.getElementById('forum');
            forum.innerHTML = ''; // Clear existing forum content.
            posts.forEach(post => {
                const postElement = document.createElement('div');
                postElement.className = 'post';
                let responsesHtml = post.responses.map((response, index) => `
                    <div class="response">
                        <span class="response-id">#${response.id}</span>
                        ${response.content}
                    </div>
                `).join('');
                postElement.innerHTML = `
                    <h2>${post.title} <span class="post-
                    id">#${post.id}</span></h2>
                    <p>${post.content}</p>
                    <button onclick="toggleResponses(this)">Show all
                    replies</button>
                    <div class="responses">
                        ${responsesHtml}
                    </div>
                `;
                forum.appendChild(postElement);
            });
            adjustResponsesVisibility(); // Call this after posts are loaded
            // to ensure correct initial visibility.
            setupImageToggle(); // Ensure images are set up for toggling after
            // dynamic content is loaded.
        })
        .catch(error => console.error('Error loading posts:', error));
}
```



# script.js

```
}

// Adjusts visibility of responses beyond the fifth for each post.
function adjustResponsesVisibility() {
  document.querySelectorAll('.post').forEach(post => {
    const responses = post.querySelectorAll('.response');
    responses.forEach((response, index) => {
      if (index >= 5) {
        response.style.display = 'none'; // Hide responses beyond the
        fifth.
      }
    });
  });
}

// Toggles the visibility of responses for a post and updates button text.
function toggleResponses(button) {
  const responses = button.nextElementSibling.querySelectorAll('.response');
  let isAnyHidden = Array.from(responses).slice(5).some(response =>
  response.style.display === 'none');

  responses.forEach((response, index) => {
    if (index >= 5) {
      response.style.display = isAnyHidden ? 'block' : 'none'; // Toggle
      display for responses beyond the fifth.
    }
  });

  button.textContent = isAnyHidden ? "Hide replies" : "Show all replies"; //
  Update button text.
}

// Sets up click event listeners on images to toggle their size.
function setupImageToggle() {
  document.querySelectorAll('.post-image, .response-image').forEach(image =>
  {
    // initially set to 150px
    image.style.maxWidth = '150px';

    image.addEventListener('click', function() {
      if (this.style.width === '150px') {
        // Restore original size
        this.style.maxWidth = '500px';
      } else {
        // Set to 150px
      }
    });
  });
}
```

# script.js

```
        this.style.maxWidth = '150px';
    }
    });
});
}

let formVisible = false;

// Show or hide create thread form.
function toggleThreadForm() {
    if (!formVisible) {
        formVisible = true;
        document.getElementById("toggleCreateThread").style.visibility =
"visible";
        document.getElementById("toggleCreateThreadButton").innerText = "Fjal
Tráð Form";
    } else {
        formVisible = false;
        document.getElementById("toggleCreateThread").style.visibility =
"hidden";
        document.getElementById("toggleCreateThreadButton").innerText =
"Vís Tráð Form";
    }
}

let replyVisible = false;

// Show or hide reply form.
function toggleReplyForm() {
    if (!replyVisible) {
        replyVisible = true;
        document.getElementById("toggleReplyForm").style.visibility =
"visible";
        document.getElementById("toggleCreateReplyButton").innerText = "Fjal
Svara Tráð";
    } else {
        replyVisible = false;
        document.getElementById("toggleReplyForm").style.visibility =
"hidden";
        document.getElementById("toggleCreateReplyButton").innerText = "Svara
Tráð";
    }
}

// Function to navigate back based on the current page context
```

## script.js

```
function goBackStructured() {  
    const currentURL = window.location.href;  
    const urlSearchParams = new URLSearchParams(currentURL.split('?')[1]);  
    const forumId = urlSearchParams.get("forum_id");  
  
    // Navigate to the determined URL  
    window.location.href = "view_forum.php?forum_id=" + forumId;  
}
```

# style.css

```
.gradient-box {
  height: 100px;
  width: 100%;          /* Full width */
  background: linear-gradient(to bottom, yellow 0%, orange 100%);
}

.small-title {
  padding-left: 10px;
  padding-top: 10px;
  font: 1.5em Arial, "Helvetica Neue", Helvetica, sans-serif;
}

.title {
  padding-left: 20px;
  font: 2.5em Arial, "Helvetica Neue", Helvetica, sans-serif;
}

/*Home page*/
.home {
  margin: 20px auto;
  max-width: 800px;
  padding: 20px;
  background-color: #FAFFFB;
  border-radius: 10px;
  box-shadow: 0 4px 8px rgba(0,0,0,0.1);
}

.forum h1 {
  text-align: center;
}

.forum-link-container {
  margin-bottom: 20px; /* Spacing before the rest of the forum content */
  text-align: center; /* Center the link */
}

.forum-link {
  display: inline-block; /* Allows setting padding and margins */
  padding: 10px 20px; /* Padding around the link text */
  background-color: #B7E4C7; /* Match the button color for consistency */
  color: #333; /* Text color */
  border-radius: 15px; /* Rounded corners */
  text-decoration: none; /* Remove underline from the link */
  font-size: 1.2rem; /* Larger text for the link */
  font-weight: bold; /* Make link text bold */
  box-shadow: 0 2px 4px rgba(0,0,0,0.1); /* Subtle shadow for depth */
}
```

# style.css

```
}

.forum-link:hover {
    background-color: #9FD3B8; /* Darker shade on hover for interaction
feedback */
}

body {
    font-family: Arial, sans-serif;
    background-color: #E8F0F2;
    color: #4A4A4A;
}

.forum {
    margin: 20px auto;
    max-width: 800px;
    padding: 20px;
    background-color: #FAFFFB;
    border-radius: 10px;
    box-shadow: 0 4px 8px rgba(0,0,0,0.1);
}

#make-post-form {
    padding-bottom: 10px;
}

nav {
    margin: 20px auto;
    max-width: 800px;
    padding: 20px;
    background-color: #FAFFFB;
    border-radius: 10px;
    box-shadow: 0 4px 8px rgba(0,0,0,0.1);
}

textarea, input[type="file"] {
    width: 50%;
    display: block; /* Ensure block-level element behavior for proper width
application */
    margin: 0 auto 10px; /* Center align the elements and add margin at the
bottom */
    padding: 10px;
```

# style.css

```
border: 1px solid #ccc;
border-radius: 5px;
}

textarea {
  min-height: 100px;
}

button[type="submit"] {
  display: block; /* Make button block-level to allow margin auto for
centering */
  width: fit-content; /* Adjust width to fit content */
  margin: 0 auto; /* Center the button */
  background-color: #B7E4C7;
  color: #333;
  padding: 10px 20px;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}

button[type="submit"]:hover {
  background-color: #9FD3B8;
}

.post {
  margin-bottom: 20px;
  padding: 20px;
  background-color: #FFFFFF;
  border: 1px solid #DDEDEE;
  border-radius: 8px;
}

.post-id, .response-id {
  font-size: 0.8rem;
  color: #6c757d;
  margin-left: 10px;
}

.post-image, .response-image {
  /* width: 150px;
height: 150px; */
  object-fit: cover;
  cursor: pointer;
```

# style.css

```
margin-right: 10px;
float: left;
}

.post, .response {
  overflow: hidden;
  padding-left: 10px;
  max-width: 810px;
  margin-left: auto;
  margin-right: auto;
  margin-bottom: 10px;
}

.response {
  margin-top: 10px;
  padding: 10px;
  background-color: #ECFFE4;
  border-left: 4px solid #B7E4C7;
  border-radius: 5px;
}

.hidden {
  display: none;
}

button {
  background-color: #B7E4C7;
  color: #333;
  border: none;
  padding: 8px 16px;
  border-radius: 5px;
  cursor: pointer;
  font-size: 1rem;
  transition: background-color 0.3s;
}

button:hover {
  background-color: #9FD3B8;
}

/*Home page*/
.home {
  margin: 20px auto;
  max-width: 800px;
  padding: 20px;
  background-color: #FAFFB;
}
```

# style.css

```
border-radius: 10px;
box-shadow: 0 4px 8px rgba(0,0,0,0.1);
}

.forum h1 {
  text-align: center;
}

.forum-link-container {
  margin-bottom: 20px; /* Spacing before the rest of the forum content */
  /* text-align: center; Center the link */
}

.forum-link {
  display: inline-block; /* Allows setting padding and margins */
  padding: 10px 20px; /* Padding around the link text */
  background-color: #B7E4C7; /* Match the button color for consistency */
  color: #333; /* Text color */
  border-radius: 15px; /* Rounded corners */
  text-decoration: none; /* Remove underline from the link */
  font-size: 1.2rem; /* Larger text for the link */
  font-weight: bold; /* Make link text bold */
  box-shadow: 0 2px 4px rgba(0,0,0,0.1); /* Subtle shadow for depth */
}

.forum-link:hover {
  background-color: #9FD3B8; /* Darker shade on hover for interaction
feedback */
}

body {
  font-family: Arial, sans-serif;
  background-color: #E8F0F2;
  color: #4A4A4A;
}

.forum {
  margin: 20px auto;
  max-width: 800px;
  padding: 20px;
  background-color: #FAFFFB;
  border-radius: 10px;
  box-shadow: 0 4px 8px rgba(0,0,0,0.1);
}
```



# style.css

```
}

.thread {
  margin: 5px auto;
  max-width: 820px;
  padding: 10px;
  background-color: #FAFFFB;
  border-radius: 10px;
  box-shadow: 0 4px 8px rgba(0,0,0,0.1);
}

.button-form {
  margin: 5px auto;
  max-width: 820px;
  padding: 10px;
}

#toggleCreateThread {
  visibility: hidden;
}

#toggleReplyForm {
  visibility: hidden;
}

form {
  margin: 20px auto;
  max-width: 800px;
  padding: 20px;
  background-color: #FAFFFB;
  border-radius: 10px;
  box-shadow: 0 4px 8px rgba(0,0,0,0.1);
}

#make-post-form {
  padding-bottom: 10px;
}

textarea, input[type="file"] {
  width: 50%;
  display: block; /* Ensure block-level element behavior for proper width
application */
```

# style.css

```
    margin: 0 auto 10px; /* Center align the elements and add margin at the
bottom */
    padding: 10px;
    border: 1px solid #ccc;
    border-radius: 5px;
}

textarea {
    min-height: 100px;
}

button[type="submit"] {
    display: block; /* Make button block-level to allow margin auto for
centering */
    width: fit-content; /* Adjust width to fit content */
    margin: 0 auto; /* Center the button */
    background-color: #B7E4C7;
    color: #333;
    padding: 10px 20px;
    border: none;
    border-radius: 5px;
    cursor: pointer;
}

button[type="submit"]:hover {
    background-color: #9FD3B8;
}

.response {
    margin-top: 10px;
    padding: 10px;
    background-color: #ECFEEF;
    border-left: 4px solid #B7E4C7;
    border-radius: 5px;
}

.hidden {
    display: none;
}
```

## view\_forum.php

```
<?php
include 'conn.php'; // Include your database connection

$forum_id = isset($_GET['forum_id']) ? (int)$_GET['forum_id'] : 0;
$page = isset($_GET['page']) ? (int)$_GET['page'] : 1;
$threadsPerPage = 10;
$offset = ($page - 1) * $threadsPerPage;

try {
    $sql = "SELECT thread_id, thread_title, created_at FROM kjak_thread WHERE
forum_id = :forum_id ORDER BY created_at DESC LIMIT :offset, :threadsPerPage";
    $stmt = $pdo->prepare($sql);
    $stmt->bindParam(':forum_id', $forum_id, PDO::PARAM_INT);
    $stmt->bindParam(':offset', $offset, PDO::PARAM_INT);
    $stmt->bindParam(':threadsPerPage', $threadsPerPage, PDO::PARAM_INT);
    $stmt->execute();
    $threads = $stmt->fetchAll();

    // Count total threads for pagination
    $sqlTotal = "SELECT COUNT(*) FROM kjak_thread WHERE forum_id = :forum_id";
    $stmtTotal = $pdo->prepare($sqlTotal);
    $stmtTotal->bindParam(':forum_id', $forum_id, PDO::PARAM_INT);
    $stmtTotal->execute();
    $totalThreads = $stmtTotal->fetchColumn();
    $totalPages = ceil($totalThreads / $threadsPerPage);
} catch (Exception $e) {
    die("Error: " . $e->getMessage());
}

// Only check for forum name if it is not in the URL.
if ($forum_name == '') {
    try {
        $sqlForum = "SELECT forum_name FROM kjak_table WHERE forum_id =
:forum_id";
        $stmtForum = $pdo->prepare($sqlForum);
        $stmtForum->execute(['forum_id' => $forum_id]);
        $forum_name = $stmtForum->fetchColumn();
    } catch (Exception $e) {
        die("Error fetching forum name: " . $e->getMessage());
    }
}

// Check if the form has been submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $pdo->beginTransaction(); // Start transaction
```

## view\_forum.php

```
try {
    // Extract and sanitize input
    $forum_id = isset($_POST['forum_id']) ? (int)$_POST['forum_id'] : 0;
    $thread_title = filter_input(INPUT_POST, 'thread_title',
FILTER_SANITIZE_STRING);
    $author_name = filter_input(INPUT_POST, 'author_name',
FILTER_SANITIZE_STRING) ?? 'Anonymous'; // Default to 'Anonymous' if empty
    $post_text = filter_input(INPUT_POST, 'post_text',
FILTER_SANITIZE_STRING);
    $created_at = date('Y-m-d H:i:s');

    // Insert new thread
    $sqlThread = "INSERT INTO kjak_thread (forum_id, thread_title,
created_at) VALUES (:forum_id, :thread_title, :created_at)";
    $stmtThread = $pdo->prepare($sqlThread);
    $stmtThread->execute(['forum_id' => $forum_id, 'thread_title' =>
$thread_title, 'created_at' => $created_at]);
    $thread_id = $pdo->lastInsertId();

    // Insert opening post without image first
    $sqlPost = "INSERT INTO kjak_post (thread_id, author_name, post_text,
created_at) VALUES (:thread_id, :author_name, :post_text, :created_at)";
    $stmtPost = $pdo->prepare($sqlPost);
    $stmtPost->execute(['thread_id' => $thread_id, 'author_name' =>
$author_name, 'post_text' => $post_text, 'created_at' => $created_at]);
    $post_id = $pdo->lastInsertId();

    // Handle file upload
    if (!empty($_FILES["post_image"]["name"])) {
        $target_dir = "uploads/";
        $fileExtension = pathinfo($_FILES["post_image"]["name"],
PATHINFO_EXTENSION);
        // Extract the original filename without the extension
        $originalFilename = pathinfo($_FILES["post_image"]["name"],
PATHINFO_FILENAME);
        // Sanitize the filename to remove spaces and special characters
        $safeFilename = preg_replace("/[^a-zA-Z0-9_-]/", "",
$originalFilename);
        // Construct the new filename including the directory, the
sanitized original filename, the post ID, and the file extension
        $newFileName = $safeFilename . "_" . $post_id . "." .
$fileExtension;
        $target_file = $target_dir . $newFileName;
```

## view\_forum.php

```
        if (move_uploaded_file($_FILES["post_image"]["tmp_name"],
$target_file)) {
            // Successfully uploaded, update post record with new image
            path
            $sqlUpdatePost = "UPDATE kjak_post SET post_image =
:post_image WHERE post_id = :post_id";
            $stmtUpdatePost = $pdo->prepare($sqlUpdatePost);
            $stmtUpdatePost->execute(['post_image' => $target_file,
'post_id' => $post_id]);
        } else {
            throw new Exception("Failed to upload file.");
        }
    }

    $pdo->commit(); // Commit transaction
    header("Location: view_thread.php?thread_id=" . $thread_id .
"&forum_id=" . $forum_id); // Redirect to the newly created thread
    exit();
} catch (Exception $e) {
    $pdo->rollback(); // Rollback transaction on error
    echo "Error: " . $e->getMessage();
}
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" type="text/css" href="style.css">
    <title><?php echo htmlspecialchars($forum_name); ?></title>
</head>
<div class="nav-bar">
    <!-- "Go Back" Button -->
    <button onclick="window.location.href='index.php'">Go Back</button>
</div>
<body>
    <div id="forum" class="forum">
        <div class="small-title">Forum:</div>
        <div class="title"><?php echo htmlspecialchars($forum_name); ?></div>
    </div>
    <?php foreach ($threads as $thread): ?>
        <div class="thread">
```

## view\_forum.php

```
<a href="view_thread.php?thread_id=<?php echo
htmlspecialchars($thread['thread_id']); ?>&forum_id=<?php echo
htmlspecialchars($forum_id); ?>">
    <?php echo htmlspecialchars($thread['thread_title']); ?>
</a>
- <?php echo htmlspecialchars($thread['created_at']); ?>
</div>
<?php endforeach; ?>

<?php if ($totalPages > 1): ?>
    <nav>
        Síða:
        <?php for ($i = 1; $i <= $totalPages; $i++): ?>
            <a href="?forum_id=<?php echo $forum_id; ?>&page=<?php echo
$i; ?>"><?php echo $i; ?></a>
        <?php endforeach; ?>
    </nav>
<?php endif; ?>

<!-- <a href="create_thread.php?forum_id=<?php echo $forum_id;
?>&forum_name=<?php echo urlencode($forum_name); ?>">Nýggjan Tráð</a> -->

<div class="button-form">
    <button id="toggleCreateThreadButton"
onClick="toggleThreadForm();">Vís Tráð Form</button>
</div>
<form id="toggleCreateThread" style="visibility: hidden;"
action="create_thread.php" method="post" enctype="multipart/form-data">
    <input type="hidden" name="forum_id" value="<?php echo
$_GET['forum_id']; ?>">
    <label for="thread_title">Tráð Navn:</label>
    <input type="text" id="thread_title" name="thread_title"
required><br><br>
    <label for="author_name">Títt Navn (valfrítt):</label>
    <input type="text" id="author_name" name="author_name"><br><br>
    <label for="post_text">Tekstur: </label>
    <textarea id="post_text" name="post_text" required></textarea><br><br>
    <label for="post_image">Viðheft mynd (valfrítt):</label>
    <input type="file" id="post_image" name="post_image"><br><br>
    <input type="submit" value="Stovna Tráð">
</form>
<script src="script.js"></script>
</body>
</html>
```

view\_forum.php

# view\_thread.php

```
<?php
include 'conn.php'; // Include your database connection

$thread_id = isset($_GET['thread_id']) ? (int)$_GET['thread_id'] : 0;
$forum_id = $_GET['forum_id'];

// Fetch thread details (optional, if you want to display thread title or
other info)
try {
    $sqlThread = "SELECT thread_title FROM kjak_thread WHERE thread_id =
:thread_id";
    $stmtThread = $pdo->prepare($sqlThread);
    $stmtThread->execute(['thread_id' => $thread_id]);
    $thread = $stmtThread->fetch();
} catch (Exception $e) {
    die("Error: " . $e->getMessage());
}

// Define number of posts per page
$postsWithPage = 10;

// Calculate total number of posts in this thread
$sqlCountPosts = "SELECT COUNT(*) FROM kjak_post WHERE thread_id =
:thread_id";
$stmtCountPosts = $pdo->prepare($sqlCountPosts);
$stmtCountPosts->execute(['thread_id' => $thread_id]);
$totalPosts = $stmtCountPosts->fetchColumn();

// Calculate total pages
$totalPages = ceil($totalPosts / $postsPerPage);

// Determine current page
$page = isset($_GET['page']) ? (int)$_GET['page'] : $totalPages; // Default to
last page

// Calculate offset
$offset = ($page - 1) * $postsPerPage;

// Fetch posts for the thread
try {
    $sqlPosts = "SELECT post_id, author_name, post_text, post_image,
created_at FROM kjak_post WHERE thread_id = :thread_id ORDER BY created_at ASC
LIMIT :offset, :postsPerPage";
    $stmtPosts = $pdo->prepare($sqlPosts);
    $stmtPosts->bindParam(':thread_id', $thread_id, PDO::PARAM_INT);
```



# view\_thread.php

```
$stmtPosts->bindParam(':offset', $offset, PDO::PARAM_INT);
$stmtPosts->bindParam(':postsPerPage', $postsPerPage, PDO::PARAM_INT);
$stmtPosts->execute();
$posts = $stmtPosts->fetchAll();
} catch (Exception $e) {
    die("Error: " . $e->getMessage());
}

// Check if the form has been submitted to add a new post
if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['submit_post'])) {
    $author_name = $_POST['author_name'] ?? 'Anonymous';
    $post_text = $_POST['post_text'];
    $post_image = ''; // Implement image upload logic here
    $created_at = date('Y-m-d H:i:s');

    try {
        // Begin transaction
        $pdo->beginTransaction();

        // Insert new post without image first
        $sqlInsertPost = "INSERT INTO kjak_post (thread_id, author_name,
post_text, created_at) VALUES (:thread_id, :author_name, :post_text,
:created_at)";
        $stmtInsertPost = $pdo->prepare($sqlInsertPost);
        $stmtInsertPost->execute(['thread_id' => $thread_id, 'author_name' =>
$author_name, 'post_text' => $post_text, 'created_at' => $created_at]);
        $post_id = $pdo->lastInsertId(); // Get the ID of the newly created
post

        // Handle file upload
        if (isset($_FILES["post_image"]["name"]) &&
$_FILES["post_image"]["name"] != '') {
            $target_dir = "uploads/";
            $fileExtension = pathinfo($_FILES["post_image"]["name"],
PATHINFO_EXTENSION);
            // Extract the original filename without the extension
            $originalFilename = pathinfo($_FILES["post_image"]["name"],
PATHINFO_FILENAME);
            // Sanitize the filename to remove spaces and special characters
            $safeFilename = preg_replace("/[^a-zA-Z0-9_-]/", "",
$originalFilename);
            // Construct the new filename including the directory, the
sanitized original filename, the post ID, and the file extension
            $newFileName = $safeFilename . "_" . $post_id . "." .
$fileExtension;
```

# view\_thread.php

```
        $target_file = $target_dir . $newFileName;

        if (move_uploaded_file($_FILES["post_image"]["tmp_name"],
$target_file)) {
            // Update the post with the new image filename
            $sqlUpdatePost = "UPDATE kjak_post SET post_image =
:post_image WHERE post_id = :post_id";
            $stmtUpdatePost = $pdo->prepare($sqlUpdatePost);
            $stmtUpdatePost->execute(['post_image' => $target_file,
'post_id' => $post_id]);
        }
    }
    // what posts should be shown.
    $newPage = ceil(($totalPosts + 1) / $postsPerPage);

    // Commit transaction
    $pdo->commit();

    header("Location: view_thread.php?thread_id=" . $thread_id .
"&forum_id=" . $_GET['forum_id'] . "&page=" . $newPage); // Redirect to the
last page with the new post
    exit();
} catch (Exception $e) {
    $pdo->rollback();
    echo "Error adding post: " . $e->getMessage();
}
}

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title><?php echo htmlspecialchars($thread['thread_title'] ?? 'Thread');
?></title>
    <link rel="stylesheet" type="text/css" href="style.css">
</head>
<div class="nav-bar">
    <!-- "Go Back" Button -->
    <button onclick="goBackStructured()">Go Back</button>
</div>
<body>
    <div id="forum" class="forum">
```

# view\_thread.php

```
<div class="small-title">Tráður:</div>
<div class="title"><?php echo
htmlspecialchars($thread['thread_title']); ?></div>
</div>

<?php foreach ($posts as $post): ?>
    <div class="post">
        <p><?php if ($post['post_image']): ?>
            
            <?php endif; ?>
            <h2><?php echo htmlspecialchars($post['author_name']); ?></h2>
            <p><?php echo nl2br(htmlspecialchars($post['post_text']));
?></p>

            </p>
            <small>Skríva á: <?php echo htmlspecialchars($post['created_at']);
?></small>
        </div>
    <?php endforeach; ?>

<nav>
    <div>
        Síða:
        <?php for ($i = 1; $i <= $totalPages; $i++): ?>
            <a href="?thread_id=<?php echo $thread_id; ?>&forum_id=<?php echo
htmlspecialchars($forum_id); ?>&page=<?php echo $i; ?>"><?php echo $i; ?></a>
            <?php endfor; ?>
        </div>
    </nav>

    <div class="button-form">
        <button id="toggleCreateReplyButton"
onClick="toggleReplyForm();">Svara tráð</button>
    </div>

    <form id="toggleReplyForm" style="visibility: hidden;"
action="view_thread.php?thread_id=<?php echo $thread_id; ?>&forum_id=<?php
echo htmlspecialchars($forum_id); ?>" method="post" enctype="multipart/form-
data">
        <label for="author_name">Títt navn (valfrítt):</label>
        <input type="text" id="author_name" name="author_name"><br><br>
        <label for="post_text">Títt svar:</label>
        <textarea id="post_text" name="post_text" required></textarea><br><br>
        <input type="file" id="post_image" name="post_image"><br><br>
```

## view\_thread.php

```
<input type="submit" name="submit_post" value="Post Reply">
</form>
<script src="script.js"></script>
</body>
</html>
```