

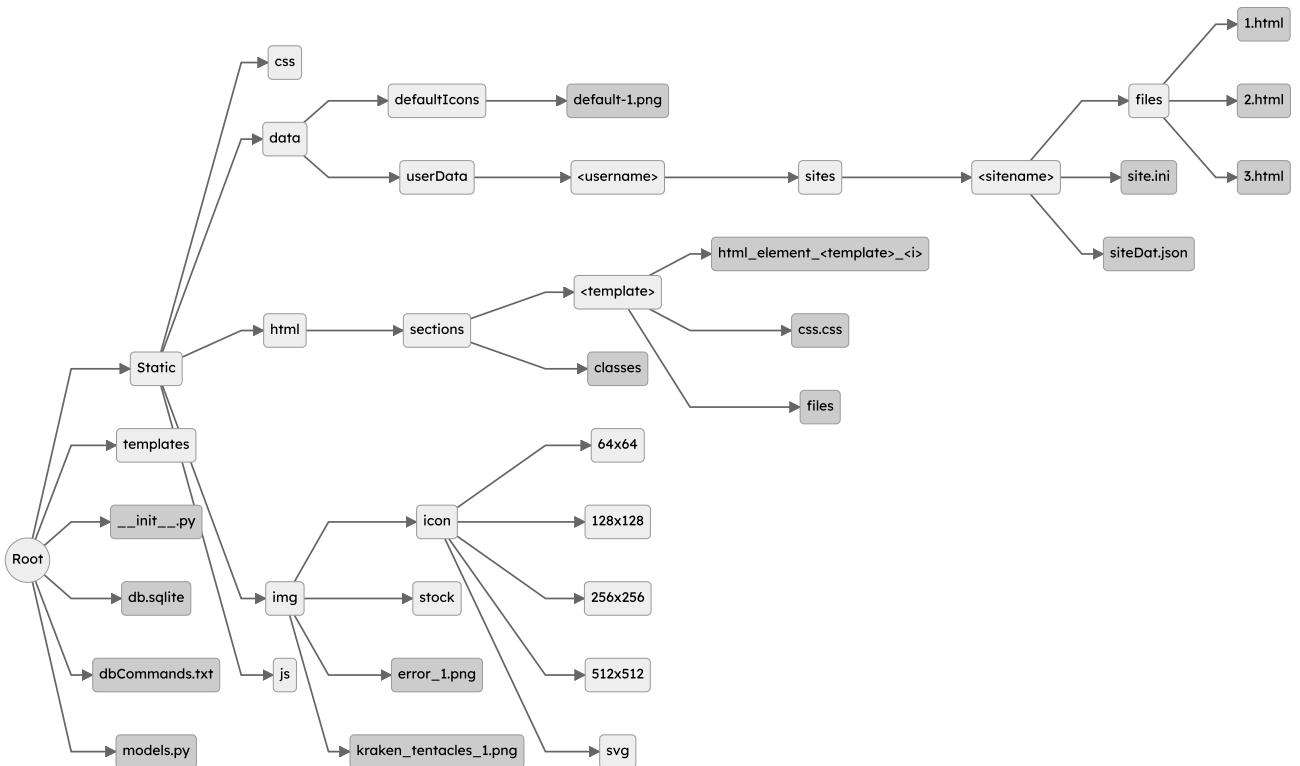
Table of Contents - Code Appendix

• File Structure Diagram	95
• Code	96
○ Root Directory	96
■ <code>_init_.py</code>	96
■ <code>dbCommands.txt</code>	107
■ <code>models.py</code>	108
■ <code>/static</code>	109
■ <code>/css</code>	109
■ <code>auth.css</code>	109
■ <code>blank.css</code>	112
■ <code>build.css</code>	114
■ <code>default_content_style.css</code>	116
■ <code>home.css</code>	116
■ <code>index.css</code>	118
■ <code>main.css</code>	120
■ <code>settings.css</code>	121
■ <code>site-create.css</code>	125
■ <code>site-edit.css</code>	131
■ <code>/html/sections</code>	135
■ <code>classes</code>	135
■ <code>/headline</code>	135
■ <code>css.css</code>	135
■ <code>files</code>	138
■ <code>html_element_headline_1.html</code>	138
■ <code>/js</code>	138
■ <code>auth.js</code>	138
■ <code>colorConversion.js</code>	140
■ <code>globalnav-floating-options.js</code>	142
■ <code>login.js</code>	143

■ main.js	143
■ signup.js	144
■ site-create-options-1.js	145
■ site-create-options-2.js	151
■ site-create-options-3.js	152
■ site-create.js	154
■ site-edit.js	157
■ /templates	162
■ base.html	162
■ home-nosite.html	165
■ home-sites.html	166
■ login.html	167
■ settings-admin.html	168
■ settings-base.html	171
■ settings-code.html	173
■ settings-dev.html	174
■ settings-looks.html	174
■ settings-profile.html	175
■ settings-sites.html	176
■ signup.html	178
■ site-create-base.html	179
■ site-create-options-1.html	180
■ site-create-options-2.html	183
■ site-create-options-3.html	185
■ site-create.html	187
■ site-edit-home.html	189
■ site-edit.html	190

File Structure Diagram

For clarity, directories which contain large amounts of files, such as `/templates` or `/static/css` do not contain those files in the diagram. Files are tinted slightly darker than directories.



Code

Root Directory/

init.py

```
# TODO: Fix color picker for site style generation page
# TODO: Add click listener to sections in dropdown
# TODO: make sure you dont render a random site that doesnt exist (line 83)

from flask import Flask, render_template, Blueprint, redirect, url_for, request,
flash, session
from flask_sqlalchemy import SQLAlchemy
from flask_login import LoginManager, login_user, login_required, current_user,
logout_user

# Flask is the application object
# render_template converts a Jinja file to html
# redirect redirects the website to another route function
# flash sends messages to the client
# request allows the code to handle form inputs
# url_for fetches the url of a server resource
# SQLAlchemy manages the SQL database
# LoginManager is the object that manages signed in users
# login_user logs in a give user

from werkzeug.security import generate_password_hash, check_password_hash

# generate_password_hash and check_password_hash are used when generating and
# authenticating users

from configparser import ConfigParser
import math

# ConfigParser is used when read/writing the user .ini files
# math.floor is used for calculating file size

# Create the database object
databaseObject=SQLAlchemy()

# The main class of the application
class Kraken():

    # Global reference to database object
    global databaseObject

    def __init__(self,host,port):

        # Assign the database object to the local db reference
        self.db=databaseObject
```

```

# import modules and add them to the object
import os,datetime
self.os=os;self.datetime=datetime;

# Initialise the flask application
self.initFlask()

# Initialise the SQL database
self.db.init_app(self.app)

# Initialise the login manager
self.loginManager=LoginManager()
# set which function routes to the login page
self.loginManager.login_view="auth_login"
self.loginManager.init_app(self.app)

# Import the User and Site entities from models.py
from models import User, Site
self.User=User
self.Site=Site

# Fetches a row from the User table in the database
@self.loginManager.user_loader
def loadUser(user_id): return self.User.query.get(user_id)

# Run the Flask application
self.app.run(host=host,port=port)

def initFlask(self):
    # Create the Flask application and set a secret key
    self.app = Flask(__name__)
    self.app.config["SECRET_KEY"]="secret-key-goes-here"
    # Set the database file URL to /db.sqlite in the root directory
    self.app.config["SQLALCHEMY_DATABASE_URI"]="sqlite:///db.sqlite"
    self.app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
    # Initialise the website pages
    self.initPages()

def initPages(self):
    # generate all of the website pages from the jinja2 html files in the
    # /templates folder
    # jinja2 is a way of programatically generating html files using variables,
    # iteration, and templates
    self.initPages_main()
    self.initPages_auth()
    self.initPages_site_create()
    self.initPages_settings()

    @self.app.route("/<name>/<site>/")
    @self.app.route("/<name>/<site>/home/")
    @login_required
    def site_edit_home(name=None,site=None):

```

```

    flash(1)
    flash(name)
    flash(site)
    if current_user.user_id!=name: return "External view of site"
    return render_template("site-edit-home.html")

@self.app.route("/<name>/<site>/edit/")
def site_edit_app(name=None,site=None):
    flash(2)
    flash(name)
    flash(site)
    return render_template("site-edit.html")

def initPages_main(self):
    @self.app.route("/")
    def main_index(): return redirect(url_for("auth_login"))

    @self.app.route("/home/")
    @login_required
    def main_home():
        # check to see if user has any sites
        if len(self.Site.query.filter_by(user_id=current_user.user_id).all()) > 0:
            flash([[x.user_id,x.name,x.private] for x in self.Site.query.filter_by(
                user_id=current_user.user_id).all()])
            return render_template("home-sites.html")
        return render_template("home-nosite.html")

    @self.app.errorhandler(404)
    @self.app.route("/404")
    def main_404(e=None): return "Page not found - i.e. you made a mistake"

    @self.app.errorhandler(500)
    @self.app.route("/500")
    def main_500(e=None): return "Server go boom - i.e. I made a mistake"

    @self.app.route("/help")
    def main_help(): return "This page dont exist yet :("

def initPages_auth(self):

    # Login page route
    @self.app.route("/login/")
    def auth_login():
        # Flash an empty list of values to stop errors in the Jinja code
        flash([False,"","","","",""])
        if current_user.is_authenticated: return redirect(url_for("main_home"))
        return render_template("login.html")

    # Login post route
    @self.app.route("/login/", methods=["post"])
    def auth_login_post():
        # Get the filled-in items from the login form
        username = request.form.get("username")

```



```

if len(verifyOutput) > 0:
    # Flash an error message and the filled in values
    flash([True, verifyOutput, name, "", username])
    return redirect(url_for("auth_signup"))

verifyOutput=self.verifyField(username, "Username", canHaveSpecialChar=False)

# Verify the username input and return an error message if invalid
if len(verifyOutput) > 0:
    # Flash an error message and the filled in values
    flash([True, verifyOutput, name, email, ""])
    return redirect(url_for("auth_signup"))

# Verify the password input and return an error message if invalid
verifyOutput=self.verifyField(password1, "Password", minLen=8)

if len(verifyOutput) > 0:
    # Flash an error message and the filled in values
    flash([True, verifyOutput, name, email, username])
    return redirect(url_for("auth_signup"))

# Return an error message if the passwords do not match
if password1!=password2:
    # Flash an error message and the filled in values
    flash([True, "Passwords do not match", name, email, username])
    return redirect(url_for("auth_signup"))

# check whether this email already has an account
if self.User.query.filter_by(email=email).first():
    flash([True, "That email is already in use", name, "", username])
    return redirect(url_for("auth_signup"))

# check whether this username already exists
user = self.User.query.filter_by(user_id=username).first()

if user:
    flash([True, "That username is already in use", name, email, ""])
    return redirect(url_for("auth_signup"))

# create a new user in the database and server-side storage
self.createUser(username, email, name, password1)

# Log in the new user and redirect them to the homepage
login_user(self.User.query.filter_by(user_id=username).first(),
remember=False)
return redirect(url_for("auth_login"))

@self.app.route("/logout/")
@self.app.route("/account/logout/")
@login_required
def auth_logout():
    logout_user()

```

```

        return redirect(url_for("auth_login"))

def initPages_site_create(self):
    @self.app.route("/home/new/")
    @login_required
    def site_create():
        out=""
        # get all current site names for the logged in user, then flash (send)
        # it to the site, where it is processed by the javascript
        for name in [x.name for x in self.Site.query.filter_by(
            user_id=current_user.user_id).all()]:
            out+=name+","
        flash(out[:-1])
        return render_template("site-create.html")

    @self.app.route("/home/new/", methods=["post"])
    @login_required
    def site_create_post():

        def listToStr(var):
            out=""
            for char in var:
                out+=char
            return out

        def replaceToDash(var):
            # replaces any invalid characters in the string var with a dash, used to
            # format the site name correctly so that there arent any errors
            var=list(var)
            for i in range(len(var)):
                if var[i] not in "qwertyuiopasdfghjklzxcvbnm-_1234567890": var[i]="-"
            return listToStr(var)

        def replaceRepeatedDashesRecursion(var):
            # recursive function to remove adjacent dashes from a string
            var=list(var)
            for i in range(len(var)):
                if i+1 >= len(var): return listToStr(var)
                if var[i] == "-" and var[i+1] == "-":
                    del var[i]
            var = list(replaceRepeatedDashesRecursion(var))
            return listToStr(var)

        # get the user inputs
        sitename = request.form.get("new_site_name")
        sitedesc = request.form.get("new_site_desc")
        isPublic = request.form.get("new_site_privacy")=="public"

        # remove adjacent dashes
        sitename=replaceRepeatedDashesRecursion(replaceToDash(sitename.lower()))

        # session can carry over variables between functions
        session["new_site_sitename"]=sitename
        session["new_site_sitedesc"]=sitedesc
        session["new_site_isPublic"]=isPublic

```

```

    return redirect(url_for("site_create_options_1"))

@self.app.route("/home/new/1")
@login_required
def site_create_options_1():
    # stop people from starting halfway through the form
    # i.e. if they didnt come from the previous site_create page,
    # send them to the start
    if not request.referrer == url_for("site_create", _external=True):
        return redirect(url_for("site_create"))
    return render_template("site-create-options-1.html")

@self.app.route("/home/new/1", methods=["post"])
@login_required
def site_create_options_1_post():
    formOutput = request.form.get("new_site_color_options_dict").split(",")
    colorOptions = {}
    for pair in formOutput:
        x=pair.split(":")
        colorOptions[x[0]]=x[1]

    session["new_site_colorOptions"]=colorOptions

    return redirect(url_for("site_create_options_2"))

@self.app.route("/home/new/2")
@login_required
def site_create_options_2():
    # stop people from starting halfway through the form
    # i.e. if they didnt come from the previous site_create page,
    # send them to the start
    if not request.referrer == url_for("site_create_options_1", _external=True):
        return redirect(url_for("site_create"))
    return render_template("site-create-options-2.html")

@self.app.route("/home/new/2", methods=["post"])
@login_required
def site_create_options_2_post():
    formOutput = request.form.get("new_site_font_face_list_active").split ","
    session["new_site_fontOptions"]=formOutput
    return redirect(url_for("site_create_options_3"))

@self.app.route("/home/new/3")
@login_required
def site_create_options_3():
    # stop people from starting halfway through the form
    # i.e. if they didnt come from the previous site_create page,
    # send them to the start
    if not request.referrer == url_for("site_create_options_2", _external=True):
        return redirect(url_for("site_create"))
    return render_template("site-create-options-3.html")

```

```

@self.app.route("/home/new/3", methods=["post"])
@login_required
def site_create_options_3_post():
    formOutput = request.form.get("new_site_style_options_list").split(",")
    styleOptions = {}
    for pair in formOutput:
        x=pair.split(":")
        if x[1] == "false": x[1]=False
        if x[1] == "true": x[1]=True
        if x[1] == "null": x[1]=None
        styleOptions[x[0]]=x[1]
    session["new_site_buttonOptions"]=styleOptions
    return redirect(url_for("site_create_generate"))

@self.app.route("/home/new/generate")
@login_required
def site_create_generate():
    if not request.referrer == url_for("site_create_options_3",_external=True):
        return redirect(url_for("site_create"))
    siteSettings={
        "name":session["new_site_sitename"],
        "user":str(current_user.user_id),
        "desc":session["new_site_sitedesc"]
        if session["new_site_sitedesc"]!="" else "No Description Set",
        "created":self.datetime.datetime.utcnow(),
        "isPublic":session["new_site_isPublic"],
        "colorOptions":session["new_site_colorOptions"],
        "fontOptions":session["new_site_fontOptions"],
        "buttonOptions":session["new_site_buttonOptions"]
    }
    self.createSiteStructure(siteSettings)

    session["new_site_sitename"]=""
    session["new_site_sitedesc"]=""
    session["new_site_isPublic"]=""
    session["new_site_colorOptions"]={}
    session["new_site_fontOptions"]=[]
    session["new_site_buttonOptions"]={}

    return redirect(url_for("site_edit_home",name=siteSettings["user"],
    site=siteSettings["name"]))

def initPages_settings(self):
    @self.app.route("/account/settings/")
    @login_required
    def settings():
        return redirect(url_for("settings_profile"))

    @self.app.route("/account/settings/profile")
    @login_required
    def settings_profile():
        flash(1,self.getUserImage(current_user.user_id))
        return render_template("settings-profile.html")

```

```

@self.app.route("/account/settings/admin")
@login_required
def settings_admin():
    flash(2)
    return render_template("settings-admin.html")

@self.app.route("/account/settings/looks")
@login_required
def settings_looks():
    flash(3)
    return render_template("settings-looks.html")

@self.app.route("/account/settings/sites")
@login_required
def settings_sites():
    flash(4)
    flash([
        [x.user_id,x.name,x.private,self.convertByteSize(
            self.getFolderSize(self.os.path.abspath(x.sitePath)))]
        for x in self.Site.query.filter_by(user_id=current_user.user_id).all()])
    return render_template("settings-sites.html")

@self.app.route("/account/settings/code")
@login_required
def settings_code():
    flash(5)
    return render_template("settings-code.html")

@self.app.route("/account/settings/dev")
@login_required
def settings_dev():
    flash(7)
    return render_template("settings-dev.html")

def createUser(self,u,e,n,p):

    # Create a new User object using the variables given
    newUser = self.User(
        user_id=u,
        name=n,
        email=e,
        password=generate_password_hash(p,method='sha256'),
        bio="",
        url="",
        archived=False,
        tabpreference=4,
    )

    # Server-side folder generation
    prefix="static/data/userData/"

```

```
        folderStructure=[self.os.path.abspath(f"{prefix}{u}"),
                         self.os.path.abspath(f"{prefix}{u}/sites/")]

        self.generateFolderStructure(folderStructure)

        self.db.session.add(newUser)
        self.db.session.commit()

    def createSiteStructure(self,siteSettings):
        sitePath=self.os.path.abspath(
            f"static/data/userData/{siteSettings['user']}/sites/{siteSettings['name']}") 
        siteConfigFile=f"{sitePath}/site.ini"

        folderStructure=[f"{sitePath}",
                         f"{sitePath}/output",
                         f"{sitePath}/files"]

        fileStructure=[siteConfigFile,
                      f"{sitePath}/siteDat.json",
                      f"{sitePath}/files/1.html"]

        self.generateFolderStructure(folderStructure)
        self.generateFileStructure(fileStructure)

        with open(f"{sitePath}/siteDat.json","w") as f:
            f.write("{\"pages\":{\"Home\":\"1.html\"},\"css\":{},\"js\":{}}")
        with open(f"{sitePath}/files/1.html","w") as f:
            f.write(self.defaultHtmlPage(siteSettings["name"],siteSettings["desc"],
                                         siteSettings["user"]))

        cfgContent=ConfigParser()
        cfgContent.read(siteConfigFile)

        section="settings"
        try: cfgContent.add_section(section)
        except: pass

        cfgContent.set(section,"name",siteSettings["name"])
        cfgContent.set(section,"user",siteSettings["user"])
        cfgContent.set(section,"desc",siteSettings["desc"])

        section="color"
        try: cfgContent.add_section(section)
        except: pass

        for key in siteSettings["colorOptions"]:
            cfgContent.set(section,key,siteSettings["colorOptions"][key])

        section="font"
        try: cfgContent.add_section(section)
        except: pass

        cfgContent.set(section,"header",siteSettings["fontOptions"][0])
```

```

cfgContent.set(section,"body",siteSettings["fontOptions"][1])

section="button"
try: cfgContent.add_section(section)
except: pass

for key in siteSettings["buttonOptions"]:
    cfgContent.set(section,key,str(siteSettings["buttonOptions"][key]).lower())

with open(siteConfigFile,"w") as f:
    cfgContent.write(f)
    f.close()

newSite = self.Site(
    name=siteSettings["name"],
    datecreated=siteSettings["created"],
    private=not siteSettings["isPublic"],
    deleted=False,
    user_id=siteSettings["user"],
    sitePath=sitePath,
)
self.db.session.add(newSite)
self.db.session.commit()

def generateFolderStructure(self,folders):
    for folder in folders:
        if self.os.path.isdir(folder): continue
        try: self.os.makedirs(folder)
        except OSError as e:
            raise OSError(
                e)

def generateFileStructure(self,files):
    for file in files:
        if self.os.path.exists(file): continue
        try:
            with open(file,"w") as f:
                if self.os.path.splitext(file)[-1] == ".json":
                    f.write("{\"content\":[]}")
                f.close()
        except OSError as e:
            raise OSError(
                e)

def getUserImage(self,u): return f"/data/userIcons/{u}.png"

def verifyField(self,field,fieldName, mustHaveChar=True,minLen=3,
    canHaveSpace=False,canHaveSpecialChar=True):
    # List of special characters for the canHaveSpecialChar flag
    specialChar="%&{}\\<>*?/$!'\\":@`|="

    # Make sure that the input given is a string, raise an exception if its not

```

```

if type(field) != str: Exception("HEY! that's not a string!")

# Check through all the flags given and throw an appropriate error message if
# input is invalid
if len(field) == 0 and mustHaveChar:
    return f"{fieldName} is not filled out."
if len(field) < minLen:
    return f"{fieldName} must be greater than {minLen-1} characters."
if not canHaveSpace and " " in field:
    return f"{fieldName} cannot contain spaces."
if not canHaveSpecialChar:
    for char in specialChar:
        if char in field:
            return f"{fieldName} cannot contain '{char}'"

return "" # Return an empty string if the input is valid

def getFolderSize(self,path):
    size=self.os.path.getsize(path)
    for sub in self.os.listdir(path):
        subPath=self.os.path.join(path,sub)
        if self.os.path.isfile(subPath): size+=self.os.path.getsize(subPath)
        elif self.os.path.isdir(subPath): size+=self.getFolderSize(subPath)
    return size

def convertByteSize(self,bytes):
    if bytes==0: return "0B"
    sizes=("B","KB","MB","GB","TB","PB","EB","ZB","YB")
    i=int(math.floor(math.log(bytes,1024)))
    p=math.pow(1024,i)
    s=round(bytes/p,2)
    if i==0: s=int(s)
    return f"{s}{sizes[i]}"

def getSiteCfg(self,siteName):
    cfgPath=self.os.path.abspath(
    "static/data/userData/{current_user.user_id}/sites/{siteName}/site.ini")

    cfgContent=ConfigParser()
    cfgContent.read(cfgPath)

    return cfgContent

def defaultHtmlPage(self,n,d,u):
    return f"""<div class=\"page\" data-content-parentview>
</div>"""

if __name__ == "__main__": Kraken("0.0.0.0",1380)

```

dbCommands.txt

```

CREATE TABLE user (
    user_id TEXT PRIMARY KEY,
    name TEXT,
    email TEXT NOT NULL,
    password TEXT NOT NULL,
    bio TEXT,
    url TEXT,
    archived BOOLEAN NOT NULL,
    tabpreference INT NOT NULL
)

CREATE TABLE site (
    site_id INTEGER PRIMARY KEY AUTOINCREMENT,
    name TEXT NOT NULL,
    datecreated DATE,
    private BOOLEAN NOT NULL,
    deleted BOOLEAN NOT NULL,
    user_id TEXT,
    sitePath TEXT,
    CONSTRAINT fk_user_id,
    FOREIGN KEY (user_id) REFERENCES user(user_id)
)

```

models.py

```

from flask_login import UserMixin

# Import the SQLAlchemy database object from the main class
from __init__ import databaseObject as db

# User class to store the user's information in the database
class User(UserMixin,db.Model):
    # Set the name of the table in the database to "user"
    __tablename__="user"

    # Define the columns in the table
    # Primary Key user_id as a string
    user_id = db.Column( db.String, primary_key=True )
    # Name as a string
    name = db.Column( db.String )
    # Email as a string, cannot be null and must be unique
    email = db.Column( db.String, nullable=False, unique=True )
    # Password as a string, cannot be null
    password = db.Column( db.String, nullable=False )
    # Bio as a text field
    bio = db.Column( db.Text )
    # URL as a text field
    url = db.Column( db.Text )
    # Archived flag as a boolean, cannot be null (default False)
    archived = db.Column( db.Boolean, nullable=False )

```

```

# Tab preference as a number, cannot be null (default four)
tabpreference = db.Column( db.Float, nullable=False )

# Setup the foreign key relationship
sites = db.relationship("Site")

# Function to return the primary key
def get_id(self): return self.user_id

# Site class to store the User's sites in the database
class Site(db.Model):
    # Set the name of the table in the database to "site"
    __tablename__="site"

    # Define the columns of the table
    # Primary Key site_id as an integer
    site_id = db.Column( db.Integer, primary_key=True )
    # Foreign Key user_id as a string, referring to user_id in the User table
    user_id = db.Column( db.String, db.ForeignKey("user.user_id") )
    # Name as a string, cannot be null
    name = db.Column( db.String, nullable=False )
    # Datecreated as a datetime format
    datecreated = db.Column( db.DateTime )
    # Private flag as a boolean, cannot be null
    private = db.Column( db.Boolean, nullable=False )
    # Deleted flag as a boolean, cannot be null (default False)
    deleted = db.Column( db.Boolean, nullable=False )
    # Sitepath as a text field
    sitePath = db.Column( db.Text )

    # Function to return the primary key
    def get_id(self): return self.site_id

```

/static/css/

/static/css/auth.css

```

.application-container {
    width:100vw;
    height:100vh;
    display:flex;
    flex-direction:row;
}

.application-content {
    width:100%;
    height:100%;
    padding: 16px;
}

```

```
.application-content .text-header-container {
  margin-bottom:64px;
  display:flex;
  flex-direction:column;
  align-items: center;
}

.application-content .text-header-container .text.one {
  margin-bottom:16px;
}

.application-content .header-option {
  position:relative;
  overflow:visible;
}

.application-content .header-option::after {
  content: "";
  background-color: var(--colors-grey-500);
  width: 70%;
  height: 4px;
  border-radius: 5px;
  position: absolute;
  bottom: -5px;
  right: -8px;
  opacity:1;
  transition:
    background-color 200ms ease-in-out,
    width 200ms ease-in-out,
    height 200ms ease-in-out,
    bottom 200ms ease-in-out,
    right 200ms ease-in-out,
    opacity 200ms ease-in-out;
}

.application-content .header-option.active::after {
  background-color: var(--colors-secondary-dark);
}

.application-content .header-option:hover::after {
  background-color: var(--colors-grey-300);
  width: 100%;
  height: 100%;
  bottom: 0;
  right: 0;
  opacity: 0.2;
}

.application-content .header-option.active:hover::after {
  background-color: var(--colors-secondary);
}

.application-content .section-header-item:not(:last-child) {
```

```
margin-bottom: 16px
}

.application-content .header-options {
  width: 50%;
  display: flex;
  justify-content: space-evenly
}

.application-content .header-option {
  cursor: pointer;
}

.application-content .header-option:active {
  opacity: .8;
}

.application-content .header-option.active {
  color: var(--colors-secondary);
}

.application-content .field-container {
  display: flex;
  flex-direction: column;
  align-items: center;
}

.application-content .field-container .field-submit {
  margin-top: 32px;
  align-self: center;
}

.application-content .field-container .field-options {
  width: 360px;
  display: flex;
  flex-direction: column;
}

.application-content .field-container .field-option:not(:last-child) {
  margin-bottom: 8px
}

.application-content .field-container .field-option {
  display: flex;
  flex-direction: row;
  justify-content: space-between;
}

.application-content .field-container .field-option .field-input-container {
  display: flex;
  flex-direction: row;
}
```

```

.application-content .field-container .field-option .field-input {
  font-family: var(--font-body);
}

.application-content .field-container .field-option .field-input-container
.eye-reveal,
.application-content .field-container .field-option .field-input-container
.eye-spacer {
  width:19px;
  height:19px;
  display:flex;
  justify-content: center;
  align-items: center;
  margin-left:8px;
}

.application-content .field-container .field-option .field-input-container
.eye-reveal:active {
  color:var(--colors-secondary);
}

.application-content .field-container .field-warning {
  color:#e63832;
  margin-bottom:8px;
}

```

/static/css/blank.css

```

/* Remove all animations, transitions and smooth scroll for people that prefer
not to see them */
@media (prefers-reduced-motion: reduce) {
  html:focus-within {
    scroll-behavior: auto !important;
  }
  *,
  *::before,
  *::after {
    animation-duration: 0.01ms !important;
    animation-iteration-count: 1 !important;
    transition-duration: 0.01ms !important;
    scroll-behavior: auto !important;
  }
}
*, 
*::before,
*::after {
  box-sizing: border-box;
  margin: 0;
  padding: 0;
  color: inherit;
}

```

```
html {
  scroll-behavior: smooth;
  -ms-text-size-adjust: 100%;
  -webkit-text-size-adjust: 100%;
}

body {
  min-height: 100vh;
  width: 100vw;
  overflow-x: hidden;
  text-rendering: optimizeSpeed;
  margin: 0;
  padding: 0;
}

ol,
ul {
  list-style: none;
}

pre,
code,
address,
caption,
th,
figcaption {
  font-size: 1em;
  font-weight: normal;
  font-style: normal
}

fieldset,
iframe {
  border: 0
}

img,
picture,
.image {
  max-width: 100%;
  display: block;
}

caption,
th {
  text-align: left
}

table {
  border-collapse: collapse;
  border-spacing: 0
}

main,
summary,
details {
  display: block
}

audio,
canvas,
video,
```

```
progress {
  vertical-align: baseline
}
body,
input,
textarea,
select,
button {
  font-synthesis: none;
  -moz-font-feature-settings: 'kern';
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  direction: ltr;
  text-align: left
}
```

/static/css/build.css

```
.globalnav {
  background: var(--colors-grey-200);
  position:fixed;
}

.globalnav-floating-options {
  transform:scale(0);
  transform-origin:bottom left;
  opacity:0;
  margin:0;
  transition:transform,opacity,margin,visibility;
  transition-delay:130ms;
  transition-duration: 260ms;
  transition-timing-function: ease-in-out;
  background-color:var(--colors-grey-100);
  position:fixed;
  bottom:0;
  left:96px;
  z-index:8;
  padding:16px 8px;
  display:flex;
  flex-direction: column;
  align-items: center;
  border-radius: 10px;
  visibility:hidden;
}

.globalnav-floating-options.is-active {
  margin-bottom:16px;
  margin-left:16px;
  transform:scale(1);
  opacity:1;
  visibility:visible;
```

```
}

.globalnav-floating-option:not(:first-of-type) {
  margin-top:16px;
}

.globalnav-floating-option-content {

}

.globalnav-floating-options-backdrop {
  z-index:7;
  position:fixed;
  width:calc(100vw - 96px);
  height:100vh;
  background-color: #000;
  top:0;
  right:0;
  opacity:0;
  transition: opacity 260ms 130ms ease-in-out, visibility 260ms 130ms ease-in-out;
  visibility: hidden;
}

.globalnav-floating-options-backdrop.is-active {
  opacity:0.4;
  visibility: visible;
}

.application-container {
  width:calc(100vw - 96px);
  height:100vh;
  display:flex;
  flex-direction:row;
  margin-left:96px;
}

.application-content {
  width:100%;
  height:100%;
  padding: 16px;
}

.application-content .text-header-container {
  margin-bottom:64px;
}

.main {
  /*height:calc(100% - 79px - 64px);*/
  width:100%;
  display:flex;
  justify-content: center;
}
```

```
.main-content {  
    width:100%;  
    height:100%;  
}  
  
.main-content.thin {  
    width:60%;  
}
```

/static/css/default_content_style.css

`default_content_style.css` has not been included due to its size (3590 SLOC).

/static/css/home.css

```
:root {  
    --link-previsited-color:var(--colors-dark);  
}  
  
.application-content .text-header-container {  
    display:flex;  
    align-items: center;  
}  
  
.application-content .empty-container {  
    display:flex;  
    flex-direction: column;  
    align-items: center;  
    width:100%;  
}  
  
.application-content .empty-container .empty-image {  
    background-image: url(..../img/kraken_tentacles_one.png);  
    width: 65%;  
    height: 346px;  
    background-size: contain;  
    background-repeat: no-repeat;  
    background-position: center;  
    margin-bottom:32px;  
}  
  
.application-content .empty-container .empty-text-container {  
    display: flex;  
    flex-direction: column;  
    align-items: center;  
}  
  
.application-content .empty-container .empty-text-container .text.two {  
    position: relative;  
    overflow:visible  
}
```

```
.application-content .empty-container .empty-text-container
.text.two:hover {
  background-position:right;
  cursor:pointer;
}

.application-content .empty-container .empty-text-container
.text.two::after {
  content: "";
  background-color: var(--colors-primary-dark);
  width: 70%;
  height: 4px;
  position: absolute;
  bottom: -5px;
  right: -18px;
  opacity:1;
  border-radius: 5px;

  transition:
    background-color 200ms ease-in-out,
    width 200ms ease-in-out,
    height 200ms ease-in-out,
    bottom 200ms ease-in-out,
    right 200ms ease-in-out,
    opacity 200ms ease-in-out;
}

.application-content .empty-container .empty-text-container
.text.two:hover::after {
  background-color: var(--colors-primary);
  width: 100%;
  height: 100%;
  bottom: 0;
  right: 0;
  opacity: 0.2;
}

.site-div {
  width: 320px;
  position: relative;
  height: 180px;
  padding: 16px;
  border-radius: 1em;
  margin-left:32px;
  margin-bottom:32px;
  border: 0px solid rgba(var(--colors-dark-rgb),0);
  transition: 200ms border ease-in-out;
  cursor:pointer;
}

.site-div:hover {
```

```
border: 5px solid var(--colors-dark-rgb),1);  
}
```

/static/css/index.css

```
.globalnav {  
background: var(--colors-grey-100);  
position:static;  
}  
  
.application-container {  
width:100vw;  
height:100vh;  
display:flex;  
flex-direction:row;  
}  
  
.application-content {  
width:100%;  
height:100%;  
padding: 16px;  
}  
  
.application-content .text-header-container {  
margin-bottom:64px;  
display:flex;  
flex-direction:column;  
align-items: center;  
}  
  
.application-content .text-header-container .text.one {  
margin-bottom:16px;  
}  
  
.application-content .header-option {  
position:relative;  
overflow:visible;  
}  
  
.application-content .header-option::after {  
content: "";  
background-color: var(--colors-grey-500);  
width: 50px;  
height: 4px;  
position: absolute;  
bottom: -5px;  
right: -8px;  
opacity:1;  
transition:  
background-color 200ms ease-in-out,  
width 200ms ease-in-out,
```

```
height 200ms ease-in-out,
bottom 200ms ease-in-out,
right 200ms ease-in-out,
opacity 200ms ease-in-out;
}

.application-content .header-option.active::after {
background-color: var(--colors-secondary-dark);
}

.application-content .header-option:hover::after {
background-color: var(--colors-grey-300);
width: 54px;
height: 19px;
bottom: 0;
right: 0;
opacity: 0.2;
}

.application-content .header-option.active:hover::after {
background-color: var(--colors-secondary);
}

.application-content .section-header-item:not(:last-child) {
margin-bottom: 16px
}

.application-content .header-options {
width:50%;
display:flex;
justify-content: space-evenly
}

.application-content .header-option {
cursor:pointer;
}

.application-content .header-option:active {
opacity:.8;
}

.application-content .header-option.active {
color:var(--colors-secondary);
}

.application-content .field-container {
display: flex;
flex-direction: column;
align-items: center;
}

.application-content .field-container .field-submit {
margin-top:32px;
```

```
    align-self: center;
}

.application-content .field-container .field-options {
  width: 360px;
  display: flex;
  flex-direction: column;
}

.application-content .field-container .field-option:not(:last-child) {
  margin-bottom: 8px
}

.application-content .field-container .field-option {
  display: flex;
  flex-direction: row;
  justify-content: space-between;
}

.application-content .field-container .field-option .field-input-container {
  display: flex;
  flex-direction: row;
}

.application-content .field-container .field-option .field-input {
  font-family: var(--font-body);
}

.application-content .field-container .field-option .field-input-container
.eyereveal,
.application-content .field-container .field-option .field-input-container
.eyespacer {
  width: 19px;
  height: 19px;
  display: flex;
  justify-content: center;
  align-items: center;
  margin-left: 8px;
}

.application-content .field-container .field-option .field-input-container
.eyereveal:active {
  color: var(--colors-secondary);
}

.application-content .field-container .field-warning {
  color: #e63832;
  margin-bottom: 8px;
}
```

`main.css` has not been included due to its size (3001 SLOC). The syntax that it defines is shown below.

Global classes:

```
.notextselect  
.nopointerevents  
.visibly-hidden  
.fake  
.box  
.no-inversion
```

Positional Classes:

```
.relative  
.sticky  
.fixed  
.abs  
.static
```

Text classes:

```
.text <light|dark|primary|secondary|accent|grey-100 => grey-800> [italic]  
[bold|thin] [ellipsis] [xl|large|default-size|small] [header|jumbo]  
[lowercase|uppercase] [notextselect] [left|center|right|justify]
```

Link classes:

```
.text .link [classes for text] [disabled] [link-slide] [notformatted]  
(link-slide class requires the css variable --link-slide-width)
```

Btn classes:

```
.btn <light|dark|primary|secondary|accent> <square|rounded|pill> [slide]  
.btn.slide [from-left|from-right]
```

If slide is set, the btn must contain a span element with syntax:

```
span .btn-content .text (all text classes apply here)
```

which contains the text of the button

A span like this is recommended even if the `.slide` class is not present so you can format the text inside separately

em classes:

```
[classes for text]
```

```
/* .section-content.fixed-width will set the width to 1440px, and will set the width to 100% when the viewport width is less than 1440px */
```

`/static/css/settings.css`

```
.settings-container {  
height: calc(100vh - 32px - 188px);  
display: flex;  
}
```

```
.settings-sidebar {
  width: 400px;
  height: 100%;
  margin-left: 10px;
}

.settings-sidebar-item-icon {
  width: 46px;
  font-size: 24px;
  display: flex;
  justify-content: center;
}

.settings-sidebar-item {
  display: flex;
  align-items: center;
  justify-content: flex-start;
  border-radius: 5px;
  padding-top: 2px;
  padding-bottom: 2px;
  position: relative;
}

.settings-sidebar-item:not(:last-of-type) {
  margin-bottom: 4px;
}

.settings-sidebar-item.is-active {
  background: var(--colors-grey-200);
}

.settings-sidebar-item.is-active::before {
  content: "";
  background-color: var(--colors-primary);
  height: 100%;
  border-radius: 5px;
  position: absolute;
  top: 0;
  left: -10px;
  width: 6px;
}

.settings-sidebar-item:hover {
  background: var(--colors-grey-300);
}

.settings-sidebar-item.is-active .settings-sidebar-item-title {
  font-weight: 700;
}

.settings-sidebar-separator {
  padding-left: 8px;
```

```
padding-top: 2px;
margin-bottom: 8px;
margin-top: 24px;
width: 100%;
border-top: 2px solid var(--colors-grey-400);
}

.settings-content {
width: calc(100% - 400px - 32px);
margin-left: 32px;
}

.settings-content .settings-content-header {
border-bottom: 2px solid var(--colors-grey-400);
width: 100%;
margin-bottom: 16px;
padding-bottom: 4px;
}

.settings-content-options {
padding-bottom: 32px;
padding-left: 16px;
}

.settings-content .settings-content-option {
width: 80%;
margin-bottom: 24px;
}

.settings-content .settings-content-option-input {
padding: 5px 12px;
border-radius: 5px;
border-style: solid;
border-width: 2px;
font-family: var(--font-body);
width: 232px;
}

.settings-content textarea.settings-content-option-input {
min-width: 190px;
min-height: 28px;
max-width: 100%;
max-height: 192px;
}

.settings-content .settings-content-option-title {
margin-bottom: 6px;
}

.settings-content .settings-content-option-caption {
margin-top: 6px;
}
```

```
.settings-content .settings-content-option-image-upload {
  width:200px;
  height:200px;
  position:relative;
  cursor:pointer;
}

.settings-content .settings-content-option-image-upload-figure {
  width:200px;
  height:200px;
  background-size: 200px 200px;
  background-position: center;
  border-radius:50%
}

.settings-content .settings-content-option-image-upload::before {
  content: "Upload Image";
  width: 100%;
  height: 100%;
  background-color: rgba(var(--colors-dark-rgb),.3);
  top: 0;
  left: 0;
  opacity:0;
  position: absolute;
  border-radius: 50%;
  color: var(--colors-dark);
  font-family: var(--font-header);
  font-size: 26px;
  font-weight: 700;
  display: flex;
  justify-content: center;
  align-items: center;
  transition:opacity 100ms ease-in-out;
}

.settings-content .settings-content-option-image-upload:hover::before {
  opacity:1;
}

.settings-content .settings-content-separator {
  width:100%;
  height:0px;
  border-top:2px solid var(--colors-grey-400);
  margin-bottom:24px;
}

.settings-content .settings-content-update .settings-content-option-caption {
  margin-top:0px;
  margin-bottom:8px;
}

.settings-content .settings-content-table {
  border: 2px solid var(--colors-grey-400);
```

```
border-radius: 10px;
overflow:hidden;
max-width:780px;
}

.settings-content .settings-content-table-header {
background-color:var(--colors-grey-100);
}

.settings-content .settings-content-table-row {
padding: 16px;
}

.settings-content .settings-content-table-row:not(:last-of-type),
.settings-content .settings-content-table-header {
border-bottom: 2px solid var(--colors-grey-400);
}

.settings-content-table-row-icon,
.settings-content-table-row-title {
margin-right:6px;
}

.settings-content-table-row-settings {
float: right;
}
```

/static/css/site-create.css

```
.new-site-form {
width:100%;
margin-bottom: 32px;
}

.new-site-form .form-input-container {
padding-bottom:8px;
padding-left:16px;
}

.new-site-form.one .form-input-container.one {
display:flex;
flex-wrap:wrap;
}

.new-site-form.one .form-input-container.one .form-input-content-column {
height:84px;
display:flex;
flex-direction:column;
justify-content:space-between;
}
```

```
.new-site-form.one .form-input-container.one .form-input-content-column
.text.one {
  padding-top: 8px;
}

.new-site-form.one .form-input-container.one .form-input-content-column * {
  height:32px
}

.new-site-form.one .form-input-container.one
[data-form-input-display="inactive"].new-site-input {
  border-color:var(--colors-grey-400);
  box-shadow: none;
}

.new-site-form.one .form-input-container.one
[data-form-input-display="success"].new-site-input {
  border-color:var(--colors-success);
  box-shadow: 0px 0px 22px -8px var(--colors-success);
}

.new-site-form.one .form-input-container.one
[data-form-input-display="warning"].new-site-input {
  border-color:var(--colors-warning);
  box-shadow: 0px 0px 22px -8px var(--colors-warning);
}

.new-site-form.one .form-input-container.one
[data-form-input-display="danger"].new-site-input {
  border-color:var(--colors-danger);
  box-shadow: 0px 0px 22px -8px var(--colors-danger);
}

.new-site-form.one .form-input-container.one .message-container {
  width:100%;
}

.new-site-form.one .form-input-container.three {
  display: flex;
  justify-content: space-between;
  align-items: center;
}

.input-checkbox {
  margin-top:15px;
  margin-bottom:15px;
  width:100%;
  display:flex;
  align-items: center;
}

.input-checkbox-input {
  float: left;
}
```

```
vertical-align: middle;
margin-right:8px;
}

.input-checkbox-icon {
font-size:26px;
margin-right:8px;
}

.input-checkbox-text-container {
display:flex;
flex-direction:column;
}

.new-site-form.two .light-dark-selector-container {
width:100%;
display:flex;
justify-content:center;
}

.new-site-form.two .light-dark-selector {
display:flex;
flex-direction:column;
width:fit-content;
}

.new-site-form.two .light-dark-selector .button-container {
width:100%;
display:flex;
flex-direction:row;
justify-content:space-evenly;
}

.new-site-form.two .color-display-container .color-display {
margin-bottom:var(--margin-m);
}

.new-site-form.two .color-display-container .light-dark-display {
display:flex;
justify-content:space-around;
}

.new-site-form.two .color-display-container .grey-display {
height: 160px;
}

.new-site-form.two .color-display-container .color-single-card {
display:flex;
align-items:center;
justify-content:center;
width:160px;
height:120px;
border-radius:1em;
```

```
    box-shadow: rgb(0 0 0 / 20%) 0px 6px 40px -10px;
}

.new-site-form.two .color-display-container .color-code {
  font-family:var(--font-header);
}

.new-site-form.two .color-display-container .color-columns {
  box-shadow: rgb(0 0 0 / 10%) 2px 6px 40px -10px;
}

.new-site-form.two .color-display-container .color-columns .color-column {
  box-shadow:none;
}

.new-site-form.two .main-color-display {
  display: flex;
  justify-content: space-between;
}

.new-site-form.two .main-color-display .color-triple-card {
  display: flex;
  flex-direction: column;
  width: -webkit-fill-available;
  height: 140px;
  align-items: center;
  justify-content: center;
  border-radius: 1em;
  box-shadow: rgb(0 0 0 / 20%) 0px 6px 40px -10px;
  overflow:hidden;
  position:relative;
}

.new-site-form.two .main-color-display .color-triple-card:not(:last-of-type) {
  margin-right:24px;
}

.new-site-form.two .main-color-display .color-triple-card-main {
  height: 60%;
  width:100%;
  display: flex;
  justify-content: center;
  align-items: center;
}

.new-site-form.two .main-color-display .color-triple-card-sub-container {
  width: 100%;
  height: 40%;
  display: flex;
  flex-direction: row;
}

.new-site-form.two .main-color-display .color-triple-card-sub {
```

```
width: 50%;  
height: 100%;  
}  
  
.new-site-form.two .color-card-picker-input {  
position: absolute;  
top: 0;  
left: 0;  
-webkit-appearance: none;  
border: none;  
width: 100%;  
height: 100%;  
background: transparent;  
opacity: 0;  
cursor: pointer;  
}  
  
.sliding-input {  
-webkit-appearance: none;  
height: 100%;  
background: transparent;  
border: none;  
}  
  
.sliding-input::-webkit-slider-thumb {  
-webkit-appearance: none;  
height: 16px;  
width: 16px;  
border-radius: 50%;  
background: var(--colors-light);  
margin-top: -3px;  
box-shadow: 0px 0px 11px -3px rgb(0 0 0 / 50%);  
cursor: pointer;  
border: 1px solid var(--colors-grey-400);  
}  
  
.sliding-input::-webkit-slider-runnable-track {  
width: 60%;  
height: 9px;  
background: var(--slider-background);  
border-radius: 3px;  
cursor: pointer;  
}  
  
.input-slider-option {  
display: flex;  
flex-direction: row;  
align-items: center;  
justify-content: space-between;  
}  
  
.input-slider-option .sliding-input {  
width: 40%;
```

```
}

.color-options {
  display:flex;
  flex-direction:column;
}

.color-options .input-slider-option:not(:last-of-type) {
  margin-bottom:var(--margin-xs)
}

.input-slider-option #new_site_colors_light_dark_bounds_slider
.sliding-input::-webkit-slider-runnable-track {
  background:linear-gradient(to right, #333333, #000000);
  border-radius:3px;
}

.input-slider-option #new_site_colors_monochromatic_temperature_slider
.sliding-input::-webkit-slider-runnable-track {
  background:linear-gradient(to right, #3962D2, #ffffff, #F19C38);
  border-radius:3px;
}

.small-number-input {
  padding: 5px;
  width: 45px;
}

.input-slider-and-number {
  display:flex;
  flex-direction:row;
  width:45%;
  align-items: center;
  justify-content: flex-end;
}

.input-slider-and-number .sliding-input {
  width:-webkit-fill-available;
  margin-right:8px;
}

.new-site-form.two .submit-container {
  display:flex;
  justify-content:center;
}

.new-site-form.three .text-options {
  margin-bottom:var(--margin-m);
}

.new-site-form.three .text-option {
  display:flex;
  flex-direction:column;
```

```

width:60%;
border-radius:0.5em;
padding:8px;
background-color:transparent;
cursor:pointer;
transition:background-color 100ms ease-in-out,border-color 100ms ease-in-out;
border-width:2px;
border-style:solid;
border-color:transparent;
}

.new-site-form.three .text-option:hover {
  background-color:var(--colors-grey-100);
}

.new-site-form.three .text-option.active {
  border-color:var(--colors-primary)
}

.new-site-form.three .text-option:not(:last-of-type) {
  margin-bottom:16px;
}

.new-site-form.four .button-option {
  display: flex;
  align-items: center;
  justify-content: space-between;
  width: 335px;
  transition: opacity 200ms ease-in-out, visibility 200ms ease-in-out;
}

.new-site-form.four .button-option-container {
  height:55px;
  margin-bottom: var(--margin-s);
  display: flex;
  align-items: center;
}

```

/static/css/site-edit.css

```

.localnav.one {
  border-radius: 1em;
  margin-right: 32px;
}
.localnav.one .localnav-item.one .localnav-item-collapsible-text,
.localnav.one .localnav-item.three .localnav-item-collapsible-text {
  width: 119px;
}
.localnav.one .localnav-item.two .localnav-item-collapsible-text {
  width: 104px;
}

```

```
.localnav.one .localnav-item.four .localnav-item-collapsible-text {
  width: 134px;
}

.lightbox-mask {
  position: absolute;
  top: 0;
  left: 0;
  right: 0;
  bottom: 0;
  z-index: 299;
  width: 100%;
  height: 100%;
  background-color: rgba(0, 0, 0, .5);
  visibility: hidden;
  opacity: 0;
  transition: opacity 200ms ease-out, visibility 200ms ease-out;
}

.lightbox-mask.shown {
  visibility: visible;
  opacity: 1;
  cursor: pointer;
}

.section-selector {
  width: 80%;
  height: 80%;
  background-color: var(--colors-grey-100);
  border-radius: 1em;
  position: relative;
  display: flex;
  flex-direction: row;
  align-items: center;
  z-index: 300;
}

.section-selector-container {
  position: absolute;
  width: calc(100% - 208px);
  height: calc(100% - 32px);
  display: flex;
  -webkit-box-align: center;
  -ms-flex-align: center;
  align-items: center;
  -webkit-box-pack: center;
  -ms-flex-pack: center;
  justify-content: center;
  right: 200px;
  opacity: 0;
  transform: scale(0.9);
  visibility: hidden;
  transition:
    right 200ms ease-out,
```

```
    opacity: 200ms ease-out,
    transform: 200ms ease-out,
    visibility: 200ms ease-out;
    z-index: 300;
}
.section-selector-container.shown {
    right: 16px;
    opacity: 1;
    transform: scale(1);
    visibility: visible;
}
.section-selector-exit-btn {
    position: absolute;
    top: 16px;
    right: 16px;
    font-size: 24px;
    cursor: pointer;
}
.section-selector-sidebar {
    width: 180px;
    height: 100%;
}
.section-selector-nav {
    background-color: transparent;
    display: block;
    height: 100%;
    width: 180px;
    max-height: 100%;
    max-width: 180px;
}
.section-selector-nav .section-selector-nav-content {
    margin: 0 auto;
    position: relative;
    height: inherit;
    z-index: 102;
    margin: auto 0;
    display: flex;
    flex-direction: column;
    justify-content: center;
}
.section-selector-nav .section-selector-nav-list {
    cursor: default;
    margin: 0 -8px;
    display: -webkit-box;
    display: -ms-flexbox;
    display: flex;
    -webkit-box-pack: justify;
    -ms-flex-pack: justify;
    justify-content: space-between;
    align-items: center;
    height: 85%;
```

```
flex-direction: column;
}
.section-selector-nav-item a.link {
  position: relative;
}

.section-selector-content {
  height: 100%;
  display: flex;
 /*width: -webkit-fill-available;*/
 width: calc(100% - 182px);
 align-items: center;
 justify-content: center;
}
.section-selector-content .section-selector-list {
  max-width: 90%;
  width: 90%;
  height: 85%;
  display: flex;
  flex-direction: column;
  align-items: center;
  overflow-y: scroll;
}
.section-selector-content .section-selector-list-item {
  width: 45%;
  height: fit-content;
  display: flex;
  justify-content: center;
}
.section-selector-content .section-selector-list::-webkit-scrollbar {
  background: transparent;
}
.section-selector-content .section-selector-list::-webkit-scrollbar-thumb {
  background-color: var(--colors-grey-300);
}

.section-selector-404-error {
  font-weight: 700;
  font-family: var(--font-header);
  font-size: 32px;
  text-align: center;
  overflow: hidden;
  text-decoration: none;
  text-transform: none;
  color: var(--colors-danger) !important;
}

.application-content {
  display: flex;
  flex-direction: row;
  justify-content: space-between;
}
```

```
.site-builder {  
    width:100%;  
    scroll-behavior: auto;  
    overflow-y: scroll;  
    overscroll-behavior: auto;  
    border: 2px solid var(--colors-grey-200);  
    border-radius: 1em;  
    --scrollbar-display: none;  
}  
  
.site-builder-preview, {  
    width:100%;  
    height:100%;  
}  
  
.site-builder-preview .page {  
    width:100%;  
}
```

/static/html/sections/

The following code has been reduced down to one example headline file, and the necessary structure required. In reality, there are a lot more files that have not been included. The ellipses denotes where code has been removed.

/static/html/sections/classes

```
headline  
...  
...
```

/static/html/sections/headline/

/static/html/sections/headline/css.css

The image data string has been removed due to its size.

```
--headline--type-1 --headline-image-figure,  
--headline--type-5 --headline-image-figure,  
--headline--type-12 --headline-image-figure {  
    background-image: url(data:image/jpeg ... )  
}  
  
...  
  
.--headline {  
    width:100%;  
    --headline-m-width:100%;
```

```
display:flex;
-webkit-box-align: center;
-ms-flex-align: center;
align-items: center;
-webkit-box-pack: center;
-ms-flex-pack: center;
justify-content: center;
flex-direction: column;
align-text:center;

padding-left:32px;
padding-right:32px;
padding-top:64px;
padding-bottom:64px;

background-color:white;
background-position: 50% 50%;
background-size: cover;
}

.--headline .--headline-text-container {
max-width:75%;
width:75%;
display:flex;
flex-direction: column;
align-items: center;
}

.--headline .--headline-text {
font-size: 64px;
font-family: var(--font-header);
font-weight: 600;
width: fit-content;
overflow: hidden;
text-decoration: none;
text-transform: none;
text-align: center;
}

.--headline .--headline-subtitle {
font-size: 24px;
font-family: var(--font-header);
font-weight: 400;
width: fit-content;
overflow: hidden;
text-decoration: none;
text-transform: none;
text-align: center;
color:#aaaaaa;
}

.--headline .--headline-btn {
margin-top:16px;
```

```
}

.--headline .--headline-image {
  max-width: 75%;
  width: 75%;
  height: 122px;
  margin-bottom: 8px;
}

.--headline .--headline-image-figure {
  width: 100%;
  height: 100%;
  background-position: center 25%;
  background-size: cover;
}

[data-preview].--headline {
  padding-left:16px;
  padding-right:16px;
  padding-top:32px;
  padding-bottom:32px;
  -webkit-user-select: none;
  -moz-user-select: none;
  -ms-user-select: none;
  user-select: none;
  transition: transform 120ms ease-in-out;
}

[data-preview].--headline .--headline-text-container {
  max-width:80%;
}

[data-preview].--headline .--headline-text {
  font-size: 32px;
}

[data-preview].--headline .--headline-subtitle {
  font-size: 12px;
}

[data-preview].--headline .--headline-btn {
  display:flex;
  margin-top:8px;
  transition: background-color 999999s cubic-bezier(1,0,1,-0.15);
}

[data-preview].--headline .--headline-btn.thin {
  padding: 0.2em 0.5em;
}

[data-preview].--headline .--headline-btn.rounded {
  border-radius:0.3em;
}
```

```
[data-preview].--headline.--headline-btn .text {  
    font-size:8px;  
}
```

/static/html/sections/headline/files

```
html_element_headline_1.html  
...
```

/static/html/sections/headline/html_element_headline_1.html

```
<div class="--headline --type-1" data-preview>  
    <div class="--headline-image">  
        <figure class="--headline-image-figure"></figure>  
    </div>  
    <div class="--headline-text-container">  
        <h1 class="--headline-text">  
            Header Text!  
        </h1>  
        <h2 class="--headline-subtitle">  
            Subtitle Text!  
        </h2>  
        <div class="--headline-btn btn primary thin rounded">  
            <span class="btn-content text uppercase">Learn More</span>  
        </div>  
    </div>  
</div>
```

/static/js/

/static/js/auth.js

```
// Function called for each field to make sure it is in the correct format, takes  
// a few arguments as flags for what makes it valid  
function verifyField(field, fieldName, mustHaveChar=true, minLen=3,  
    canHaveSpace=false, canHaveSpecialChar=true, isPassword=false) {  
    // List of special characters for the canHaveSpecialChar flag  
    specialChar="%&{}\\<>*?/$!'^\":@+`|= "  
  
    // Make sure that the input given is a string  
    if (typeof field != "string") {throw new Error("HEY! that's not a string?")}  
  
    // Check through all the flags given and throw an appropriate error message  
    // if input is invalid  
    if (field.length==0 && mustHaveChar) {return `${fieldName} is not filled out.`}  
    if (field.length<minLen) {
```

```

        return `${fieldName} must be greater than ${minLen-1} characters.`
    }
    if (!canHaveSpace && field.includes(" "))
        {return `${fieldName} cannot contain spaces.`}
    if (!canHaveSpecialChar) {
        // Iterate through each character in specialChar to see if its in the input
        // I didn't use regex for this as I wanted to be able to tell the user which
        // character wasn't allowed
        var char;
        for (var i=0;i<specialChar.length;i++) {
            char=specialChar[i]
            if (field.includes(char)) {
                return `${fieldName} cannot contain '${char}'`
            }
        }
    }
    // If the given input is a password
    if (isPassword) {
        // If it doesn't match the given regular expression for password checks
        if (!field.match(/(=?.*?[A-Z])(?=.*?[a-z])(?=.*?[0-9])/
        /(?=.*?[^#?!@${%^&*- _%&{}}\\<>*?\\/$!\\\":@+`|=]).{8,}/)) {
            return `${fieldName} must contain at least 1 of each: uppercase character,
            lowercase character, number, and special character`
        }
    }
}

// Regex pattern breakdown
//   (=?.*?[A-Z]) = contains an uppercase character
//   (=?.*?[a-z]) = contains a lowercase character
//   (=?.*?[0-9]) = contains a digit
//   (=?.*?[^#?!@${%^&*- _%&{}}\\<>*?\\/$!\\\":@+`|=]) = contains a special character
//   .{8,} = has a minimum length of 8 and no upper limit

return ""
}

// Initialise the code for the all seeing eyes to enable viewing the password
function initAllSeeingEye(element,reveal) {
    // Add onclick event to given element (the eye element)
    element.addEventListener("click", e=> {
        // toggle input type of given input between password and text
        reveal.setAttribute('type',reveal.getAttribute('type') === 'password' ?
        'text' : 'password');
        // toggle the fa-eye-slash class for the eye (this sets the icon displayed)
        element.classList.toggle('fa-eye-slash');
    })
}

// Function takes a string and returns a boolean determining whether it is in a
// valid email format, using regex
function isEmail(email) {
    return email.match(/^\\w+([\\.-]?\\w+)*@\\w+([\\.-]?\\w+)*(\\.\\w{2,3})+$/)
}

```

```
// fetch warning element and disable submit bottom
warningSpan = document.querySelector(".field-container .field-warning")
document.querySelector(".field-submit").disabled = true
```

/static/js/colorConversion.js

```
// Force a number between 0 and 1
function clamp01(val) {
    return Math.min(1, Math.max(0, val));
}

// Force a hex value to have 2 characters
function pad2(c) { return c.length == 1 ? '0' + c : '' + c }

// Assumes: r, g, and b are contained in [0, 255]
// Returns: { h, s, l } in [0,1]
function rgbToHsl(r, g, b){
    r /= 255, g /= 255, b /= 255;
    var max = Math.max(r, g, b), min = Math.min(r, g, b);
    var h, s, l = (max + min) / 2;

    console.log(h,s,l)

    if(max == min){
        h = s = 0; // achromatic
    }else{
        var d = max - min;
        s = l > 0.5 ? d / (2 - max - min) : d / (max + min);
        switch(max){
            case r: h = (g - b) / d + (g < b ? 6 : 0); break;
            case g: h = (b - r) / d + 2; break;
            case b: h = (r - g) / d + 4; break;
        }
        h /= 6;
    }

    return { h: h, s: s, l: l };
}

// Assumes: h is contained in [0, 360] and s and l are contained in [0, 100]
// Returns: { r, g, b } in the set [0, 255]
function hslToRgb(h, s, l) {

    var r, g, b;

    function hue2rgb(p, q, t) {
        if(t < 0) t += 1;
        if(t > 1) t -= 1;
        if(t < 1/6) return p + (q - p) * 6 * t;
        if(t < 1/2) return q;
```

```

    if(t < 2/3) return p + (q - p) * (2/3 - t) * 6;
    return p;
}

if(s === 0) { r = g = b = 1 }
else {
    var q = l < 0.5 ? l * (1 + s) : l + s - l * s;
    var p = 2 * l - q;
    r = hue2rgb(p, q, h + 1/3);
    g = hue2rgb(p, q, h);
    b = hue2rgb(p, q, h - 1/3);
}

return { r: (r/100)*255, g: (g/100)*255, b: (b/100)*255 };
}

// Assumes r, g, and b are contained in the set [0, 255]
// Returns a 3 or 6 character hex
function rgbToHex(r, g, b) {
    var hex = [
        pad2(Math.round(r).toString(16)),
        pad2(Math.round(g).toString(16)),
        pad2(Math.round(b).toString(16)) ];
    return hex.join("");
}

// Assumes hex is a string of six alphanumeric characters.
// It can have a hashtag at the start as well
// Returns: { r, g, b } in the set [0, 255]
function hexToRgb(hex) {
    if (hex[0]=="#") { hex=hex.slice(1) }
    const r = parseInt(hex.slice(0, 2), 16);
    const g = parseInt(hex.slice(2, 4), 16);
    const b = parseInt(hex.slice(4, 6), 16);
    return { r: r, g: g, b: b };
}

// Assumes color is in a { h, s, l } format.
// Amount is a percentage for how much you decrease it from it's current value
function desaturate(color, amount) {
    amount = (amount === 0) ? 0 : (amount || 10);
    color.s -= ((amount/100)*color.s);
    return color
}

// Assumes color is in a { h, s, l } format.
// Amount is a percentage for how much you increase it from it's current value
function saturate(color, amount) {
    amount = (amount === 0) ? 0 : (amount || 10);
    color.s += (amount/100)*(100-color.s);
    return color
}

```

```

// Assumes color is in a { h, s, l } format.
function greyscale(color) { return desaturate(color,100) }

// Assumes color is in a { h, s, l } format.
// Amount is a percentage for how much you increase it from it's current value
function lighten(color, amount) {
    amount = (amount === 0) ? 0 : (amount || 10);
    color.l += (amount/100)*(100-color.l);
    return color;
}

// Assumes color is in a { h, s, l } format.
// Amount is a percentage for how much you decrease it from it's current value
function darken(color, amount) {
    amount = (amount === 0) ? 0 : (amount || 10);
    color.l -= ((amount)/100)*color.l;
    return color;
}

// r, g, and b are contained in [0, 255],
// amount is a percentage for how much you decrease it from it's current value
function brighten(r,g,b, amount) {
    amount = (amount === 0) ? 0 : (amount || 10);
    r = Math.max(0, Math.min(255, r - Math.round(255 * - (amount / 100)))); 
    g = Math.max(0, Math.min(255, g - Math.round(255 * - (amount / 100)))); 
    b = Math.max(0, Math.min(255, b - Math.round(255 * - (amount / 100)))); 
    return { r: r, g: g, b: b }
}

```

/static/js/globalnav-floating-options.js

```

document.getElementById("globalnav-hamburger").addEventListener("click",()=>{
    document.getElementById("globalnav-hamburger").classList.toggle('is-active');
    document.querySelectorAll(".globalnav-floating-options").forEach((e)=>{
        e.classList.toggle("is-active")});
    document.querySelectorAll(".globalnav-floating-options-backdrop").forEach((e)=>{
        e.classList.toggle("is-active")});
});

document.querySelectorAll(".globalnav-floating-options-backdrop").forEach((e)=>{
    e.addEventListener("click",()=>{
        document.getElementById("globalnav-hamburger").classList.remove(
            'is-active');
        document.querySelectorAll(".globalnav-floating-options").forEach(
            (e)=>{e.classList.remove("is-active")});
        document.querySelectorAll(".globalnav-floating-options-backdrop").forEach(
            (e)=>{e.classList.remove("is-active")});
    })
});

```

/static/js/login.js

```
// function called when an input is changed, to check whether all inputs are valid
function verifyAllFields() {
    if (fields["Username"].value.length < 1) {
        warningSpan.innerText = "Username is not filled out"
        document.querySelector(".field-submit").disabled = true
        return
    }

    if (fields["Password"].value.length < 1) {
        warningSpan.innerText = "Password is not filled out"
        document.querySelector(".field-submit").disabled = true
        return
    }

    warningSpan.innerText = ""
    document.querySelector(".field-submit").disabled = false
}

// Dictionary of all fields in the form
fields={
    "Username":document.querySelector(".field-option-username .field-input"),
    "Password":document.querySelector(".field-option-password .field-input")
}

initAllSeeingEye(document.querySelector(".field-option-password .eye-reveal i"),
document.querySelector(".field-option-password .field-input"))
document.querySelectorAll(".field-input").forEach(field=>
    {field.addEventListener("change",verifyAllFields)})
```

/static/js/main.js

```
function jsE(el) {
    // takes a html element object and appends the class "js" to it.
    // Used to recognise which elements have been edited by javascript
    try {
        el.classList.add("js");
    } catch(e) {
        console.log(e);
    }
}

function removeClass(e,a) { e.classList.remove(a) }
function removeClasses(e,a) { a.forEach((x)=>{ e.classList.remove(x) }) }

function addClass(e,a) { e.classList.add(a) }
function addClasses(e,a) { a.forEach((x)=>{ e.classList.add(x) }) }

$(document).ready(function () {
```

```
// reveal the page when it has been loaded
document.body.classList.remove("visibly-hidden");jsE(document.body);
});
```

/static/js/signup.js

```
// function called when an input is changed, to check whether all inputs are valid
function verifyAllFields() {
    verifyOutput=verifyField(fields["Name"].value,"Name",true,3,true,false)

    if (verifyOutput.length > 0) {
        warningSpan.innerText = verifyOutput
        document.querySelector(".field-submit").disabled = true
        return
    }

    verifyOutput=verifyField(fields["Email"].value,"Email",true,0)

    if (verifyOutput.length > 0) {
        warningSpan.innerText = verifyOutput
        document.querySelector(".field-submit").disabled = true
        return
    }

    // check for email in the correct format
    if (!isEmail(fields["Email"].value)) {
        warningSpan.innerText = "Email is not a valid email address"
        document.querySelector(".field-submit").disabled = true
        return
    }

    verifyOutput=verifyField(fields["Username"].value,"Username",true,3,false,false)

    if (verifyOutput.length > 0) {
        warningSpan.innerText = verifyOutput
        document.querySelector(".field-submit").disabled = true
        return
    }

    verifyOutput=verifyField(fields["Password"].value,"Password",true,8,false,true,
    true)

    if (verifyOutput.length > 0) {
        warningSpan.innerText = verifyOutput
        document.querySelector(".field-submit").disabled = true
        return
    }

    // Make sure passwords match
    if (fields["Password"].value!=fields["Repeat Password"].value) {
```

```

        warningSpan.innerText = "Passwords do not match"
        document.querySelector(".field-submit").disabled = true
        return
    }

    // If no errors are called, then enable the button and clear the warning message
    warningSpan.innerText = ""
    document.querySelector(".field-submit").disabled = false
}

// Dictionary of all fields in the form
fields={
    "Name":document.querySelector(".field-option-name .field-input"),
    "Email":document.querySelector(".field-option-email .field-input"),
    "Username":document.querySelector(".field-option-username .field-input"),
    "Password":document.querySelector(".field-option-password .field-input"),
    "Repeat Password":document.querySelector(
        ".field-option-password-repeat .field-input")
}

initAllSeeingEye(
    document.querySelector(".field-option-password .eye-reveal i"),
    document.querySelector(".field-option-password .field-input"))

initAllSeeingEye(
    document.querySelector(".field-option-password-repeat .eye-reveal i"),
    document.querySelector(".field-option-password-repeat .field-input"))

document.querySelectorAll(".field-input").forEach(field=>
    {field.addEventListener("change",verifyAllFields)})

```

/static/js/site-create-options-1.js

```

function toggleOptions() {
    return
    lightOptions.classList.add("visibly-hidden");
    darkOptions.classList.add("visibly-hidden");
    if (lightModeSelected) { lightOptions.classList.remove("visibly-hidden") }
    else { darkOptions.classList.remove("visibly-hidden") }
}

function setColor(k,v) { colors[k]=v }

function setDarkMode() {
    setColor["default-background", colors["dark"]];
    setColor["default-text", colors["light"]]
}

function setLightMode() {
    setColor["default-background", colors["light"]];
    setColor["default-text", colors["dark"]]
}

```

```
}

var btnLight=document.getElementById("new_site_lightModeToggle");
var btnDark=document.getElementById("new_site_darkModeToggle");
var lightModeSelected=true;

var lightOptions=document.querySelector(
  ".new-site-form.two .light-color-options");

var darkOptions=document.querySelector(
  ".new-site-form.two .dark-color-options");

var stored=document.getElementById("color-output")

var primaryColorPicker=document.getElementById(
  "new_site_colors_primary_picker");

var secondaryColorPicker=document.getElementById(
  "new_site_colors_secondary_picker");

var accentColorPicker=document.getElementById(
  "new_site_colors_accent_picker");

// the primary color should be quite visible against the selected theme.
// Dark against light etc

// the secondary color should be slightly less so, and a rotation around
// the color wheel. Red against purple etc

// the accent color shpuld be lighter than both the primary and secondary
// colors, but not very overpowering

// there will be an option to change the starting and ending lightness of
// the grey colors, and also to tint them warm or cold slightly
// you wont be able to edit the specific colors of the greys though

// the main colors will have light and dark versions that are generated
// by the code
// their lightness can be editied but the dark one cant be lighter than 10%
// darker than the original etc
// ie there will be bounds on how light and dark the colors can go

// the default background and text colors are defined by whether the document
// is set to light or dark mode
// in the ui design, the defaults can change based on a parent element having
// .style-light or .style-dark

// the light and dark colors can be changed, but must be kept within bounds
// (eg #303030 and # f0f0f0) which havent been set
// the darkest and lightest greys cannot go darker or lighter than these colors

function updateStored() {
  var out="";var keys=Object.keys(colors);
```

```
        for (var i=0; i<keys.length; i++) { out=out+keys[i]+": "+colors[keys[i]]+", " }
        out=out.slice(0,out.length-1)
        stored.value=out
    }

    function updateLightDarkVariables() {
        var val = 100-lightDarkBoundsSlider.value;

        var newColor = darken({h:0,s:0,l:100},val/12);
        newColor = hslToRgb(newColor.h,newColor.s,newColor.l);
        newColor = rgbToHex(newColor.r,newColor.b,newColor.g);
        setColor("light","#"+newColor);

        var newColor = lighten({h:0,s:0,l:0},val/8);
        newColor = hslToRgb(newColor.h,newColor.s,newColor.l);
        newColor = rgbToHex(newColor.r,newColor.b,newColor.g);
        setColor("dark","#"+newColor);

        updateStored()
    }

    function updateLightDarkDisplay() {
        colorDisplay["light"][0].style.backgroundColor=colors["light"]
        colorDisplay["light"][1].innerHTML=colors["light"]
        colorDisplay["dark"][0].style.backgroundColor=colors["dark"]
        colorDisplay["dark"][1].innerHTML=colors["dark"]

        updateStored()
    }

    function updateColorVariables() {
        var changePercent = 20

        var newColor = hexToRgb(colors["primary"])
        // color is correct
        newColor = rgbToHsl(newColor.r,newColor.g,newColor.b)
        // color is incorrect
        newColor = darken(newColor,changePercent)
        console.log(newColor)
        newColor = hslToRgb(newColor.h,newColor.s,newColor.l);
        newColor = rgbToHex(newColor.r/255,newColor.b/255,newColor.g/255);
        setColor("primary-dark",newColor)

        newColor = hexToRgb(colors["primary"])
        newColor = lighten(rgbToHsl(newColor.r,newColor.g,newColor.b),changePercent)
        newColor = hslToRgb(newColor.h,newColor.s,newColor.l);
        newColor = rgbToHex(newColor.r,newColor.b,newColor.g);
        setColor("primary-light",newColor)

        updateStored()
    }

    function updateColorDisplays() {
```

```

colorDisplay["primary"][0].style.backgroundColor=colors["primary"];
colorDisplay["primary"][1].innerHTML=colors["primary"]

colorDisplay["primary-dark"][0].style.backgroundColor
=colors["primary-dark"];

colorDisplay["primary-light"][0].style.backgroundColor
=colors["primary-light"];

colorDisplay["secondary"][0].style.backgroundColor=colors["secondary"];
colorDisplay["secondary"][1].innerHTML=colors["secondary"]

colorDisplay["secondary-dark"][0].style.backgroundColor
=colors["secondary-dark"];

colorDisplay["secondary-light"][0].style.backgroundColor
=colors["secondary-light"];

colorDisplay["accent"][0].style.backgroundColor=colors["accent"];
colorDisplay["accent"][1].innerHTML=colors["accent"]

colorDisplay["accent-dark"][0].style.backgroundColor
=colors["accent-dark"];

colorDisplay["accent-light"][0].style.backgroundColor
=colors["accent-light"];

updateStored()
}

function updateDisplays() {
  updateStored()
  updateLightDarkDisplay()
}

function updatePrimaryColorDiv() {
  setColor("primary",primaryColorPicker.value)

  updateColorVariables()
  updateColorDisplays()
}

var options = ["light","dark","primary","primary-dark","primary-light",
  "secondary","secondary-dark","secondary-light","accent","accent-dark",
  "accent-light","grey-100","grey-200","grey-300","grey-400","grey-500",
  "grey-600","grey-700","grey-800","grey-900"]

var defaultColors = {"light":"#ffffff","dark":"#000000","primary":"#e63946",
  "primary-dark":"","primary-light":"","secondary":"#457b9d","secondary-dark":"",
  "secondary-light":"","accent":"#a8dadc","accent-dark":"","accent-light":"",
  "grey-100":"#303030","grey-200":"#474747","grey-300":"#5e5e5e",
  "grey-400":"#757575","grey-500":"#8c8c8c","grey-600":"#a3a3a3",
  "grey-700":"#bababa","grey-800":"#d1d1d1","grey-900":"#e8e8e8"}

```

```
var colors = defaultColors

var queryPrefix = ".new-site-form.two .color-display-container"

var colorDisplay = {
  "light": [
    document.querySelector(
      `${queryPrefix} .light-dark-display .light-color`),
    document.querySelector(
      `${queryPrefix} .light-dark-display .light-color .color-code`)
  ],
  "dark": [
    document.querySelector(
      `${queryPrefix} .light-dark-display .dark-color`),
    document.querySelector(
      `${queryPrefix} .light-dark-display .dark-color .color-code`)
  ],
  "primary": [
    document.querySelector(
      `${queryPrefix} .main-color-display .primary-color
      .color-triple-card-main`),
    document.querySelector(
      `${queryPrefix} .main-color-display .primary-color
      .color-triple-card-main .color-code`)
  ],
  "primary-dark": [
    document.querySelector(
      `${queryPrefix} .main-color-display .primary-color
      .color-triple-card-sub.one`)
  ],
  "primary-light": [
    document.querySelector(
      `${queryPrefix} .main-color-display .primary-color
      .color-triple-card-sub.two`)
  ],
  "secondary": [
    document.querySelector(
      `${queryPrefix} .main-color-display .secondary-color
      .color-triple-card-main`),
    document.querySelector(
      `${queryPrefix} .main-color-display .secondary-color
      .color-triple-card-main .color-code`)
  ],
  "secondary-dark": [
    document.querySelector(
      `${queryPrefix} .main-color-display .secondary-color
      .color-triple-card-sub.one`)
  ],
  "secondary-light": [

```

```
document.querySelector(
  `#${queryPrefix} .main-color-display .secondary-color
  .color-triple-card-sub.two`)
],


"accent":[
  document.querySelector(
    `#${queryPrefix} .main-color-display .accent-color
    .color-triple-card-main`),
  document.querySelector(
    `#${queryPrefix} .main-color-display .accent-color
    .color-triple-card-main .color-code`)
],
"accent-dark":[
  document.querySelector(
    `#${queryPrefix} .main-color-display .accent-color
    .color-triple-card-sub.one`)
],
"accent-light":[
  document.querySelector(
    `#${queryPrefix} .main-color-display .accent-color
    .color-triple-card-sub.two`)
],



"grey-100":[
  document.querySelector(
    `#${queryPrefix} .grey-display .color-column.g100`),
  document.querySelector(
    `#${queryPrefix} .grey-display .color-column.g100 .color-code`)
],
"grey-200":[
  document.querySelector(
    `#${queryPrefix} .grey-display .color-column.g200`),
  document.querySelector(
    `#${queryPrefix} .grey-display .color-column.g200 .color-code`)
],
"grey-300":[
  document.querySelector(
    `#${queryPrefix} .grey-display .color-column.g300`),
  document.querySelector(
    `#${queryPrefix} .grey-display .color-column.g300 .color-code`)
],
"grey-400":[
  document.querySelector(
    `#${queryPrefix} .grey-display .color-column.g400`),
  document.querySelector(
    `#${queryPrefix} .grey-display .color-column.g400 .color-code`)
],
"grey-500":[
  document.querySelector(
    `#${queryPrefix} .grey-display .color-column.g500`),
  document.querySelector(
    `#${queryPrefix} .grey-display .color-column.g500 .color-code`)
]
```

```

    ],
  "grey-600": [
    document.querySelector(
      `${queryPrefix} .grey-display .color-column.g600`),
    document.querySelector(
      `${queryPrefix} .grey-display .color-column.g600 .color-code`)
  ],
  "grey-700": [
    document.querySelector(
      `${queryPrefix} .grey-display .color-column.g700`),
    document.querySelector(
      `${queryPrefix} .grey-display .color-column.g700 .color-code`)
  ],
  "grey-800": [
    document.querySelector(
      `${queryPrefix} .grey-display .color-column.g800`),
    document.querySelector(
      `${queryPrefix} .grey-display .color-column.g800 .color-code`)
  ],
  "grey-900": [
    document.querySelector(
      `${queryPrefix} .grey-display .color-column.g900`),
    document.querySelector(
      `${queryPrefix} .grey-display .color-column.g900 .color-code`)
  ],
}


```

```

var lightDarkBoundsSlider = document.getElementById(
  "new_site_colors_light_dark_bounds_slider");

lightDarkBoundsSlider.addEventListener("mouseup", ()=>{
  updateLightDarkVariables();
  updateDisplays()
})
primaryColorPicker.addEventListener("change", ()=>{
  updatePrimaryColorDiv();
  updateStored()
})

updateStored()


```

/static/js/site-create-options-2.js

```

var textOptions = document.querySelectorAll(".new-site-form.three .text-option");

textOptions.forEach((e)=>{
  e.addEventListener("click", ()=>{
    textOptions.forEach((f)=>{
      f.classList.remove("active")
      f.querySelector(".text-option-list").name="new_site_font_face_list_inactive"
    });
  })
})


```

```
    e.classList.add("active")
    e.querySelector(".text-option-list").name="new_site_font_face_list_active"
  });
});
```

/static/js/site-create-options-3.js

```
function revealOption(e) { e.style.opacity=1;e.style.visibility="visible" }

function hideFromLeft() {
  fromLeftOptions.style.opacity=0;
  fromLeftOptions.style.visibility="hidden"
}

function makeSlide() {
  document.querySelectorAll(buttonsPrefix+":not(.slide-or-static) .btn")
    .forEach((e)=>{ e.classList.add("slide") })
}

function makeStatic() {
  document.querySelectorAll(buttonsPrefix+":not(.slide-or-static) .btn")
    .forEach((e)=>{ e.classList.remove("slide") })
}

function makeSquare() {
  document.querySelectorAll(buttonsPrefix+":not(.corner-options) .btn")
    .forEach((e)=>{ removeClasses(e,[ "rounded", "pill" ]);addClass(e,"square") })
}

function makeRounded() {
  document.querySelectorAll(buttonsPrefix+":not(.corner-options) .btn")
    .forEach((e)=>{ removeClasses(e,[ "square", "pill" ]);addClass(e,"rounded") })
}

function makePill() {
  document.querySelectorAll(buttonsPrefix+":not(.corner-options) .btn")
    .forEach((e)=>{ removeClasses(e,[ "rounded", "square" ]);addClass(e,"pill") })
}

function makeThin() {
  document.querySelectorAll(buttonsPrefix+".left-or-right .btn")
    .forEach((e)=>{ e.classList.add("thin") })
}

function makeThick() {
  document.querySelectorAll(buttonsPrefix+".left-or-right .btn")
    .forEach((e)=>{ e.classList.remove("thin") })
}

function enableSubmit() {
  document.querySelector(".new-site-form.four .field-submit")
```

```
.removeAttribute("disabled")
}

function disableSubmit() {
  document.querySelector(".new-site-form.four .field-submit")
    .setAttribute("disabled","");
}

function updateStored() {
  var out="sliding:"+slidingPreference+,cornerType:+squaredPreference
  +,thin:+thinPreference+,fromLeft:+fromLeftPreference

  stored.value=out
}

var slidingPreference = null;
var squaredPreference = null
var thinPreference = null;
var fromLeftPreference = null;
var canOpenFromLeft = false;
var hasOpenFromLeft = false;

var slidingOptions = document.querySelector(
  ".new-site-form.four .button-option.slide-or-static")

var squaredOptions = document.querySelector(
  ".new-site-form.four .button-option.corner-options")

var thinOptions = document.querySelector(
  ".new-site-form.four .button-option.thin-or-large")

var fromLeftOptions = document.querySelector(
  ".new-site-form.four .button-option.left-or-right")

var buttonsPrefix = ".new-site-form.four .button-option"
var stored=document.getElementById("style-option-output")

slidingOptions.querySelector(".option-true").addEventListener("click",(e)=>{
  slidingPreference=true;
  revealOption(squaredOptions);
  makeSlide();
  canOpenFromLeft=true;
  if(hasOpenFromLeft){revealOption(fromLeftOptions)}
  if(thinPreference!=null){revealOption(fromLeftOptions)}
  if(fromLeftPreference==null){disableSubmit()}
  updateStored();
})
slidingOptions.querySelector(".option-false").addEventListener("click",(e)=>{
  slidingPreference=false;
  revealOption(squaredOptions);
  makeStatic();
  canOpenFromLeft=false;
  hideFromLeft();
})
```

```

        updateStored();
    })

squaredOptions.querySelector(".square").addEventListener("click", (e)=>{
    squaredPreference="square";
    revealOption(thinOptions);
    makeSquare();
    updateStored();
})
squaredOptions.querySelector(".rounded").addEventListener("click", (e)=>{
    squaredPreference="rounded";
    revealOption(thinOptions);
    makeRounded();
    updateStored();
})
squaredOptions.querySelector(".pill").addEventListener("click", (e)=>{
    squaredPreference="pill";
    revealOption(thinOptions);
    makePill();
    updateStored();
})

thinOptions.querySelector(".option-true").addEventListener("click", (e)=>{
    thinPreference=true;
    if(canOpenFromLeft){revealOption(fromLeftOptions);hasOpenFromLeft=true};
    makeThin();
    if(!(slidingPreference)){enableSubmit()};
    updateStored();
})
thinOptions.querySelector(".option-false").addEventListener("click", (e)=>{
    thinPreference=false;
    if(canOpenFromLeft){revealOption(fromLeftOptions);hasOpenFromLeft=true};
    makeThick();
    if(!(slidingPreference)){enableSubmit()};
    updateStored();
})

fromLeftOptions.querySelector(".option-true").addEventListener("click", (e)=>{
    fromLeftPreference=true;
    enableSubmit();
    updateStored();
})
fromLeftOptions.querySelector(".option-false").addEventListener("click", (e)=>{
    fromLeftPreference=false;
    enableSubmit();
    updateStored();
})

```

/static/js/site-create.js

```
String.prototype.replaceAt = function(index, replacement) {
    return this.substr(0, index) + replacement
    + this.substr(index + replacement.length);
}

function checkFormSubmitButton() {
    if (!(formName.getAttribute("data-form-input-display") == "success" ||
        formName.getAttribute("data-form-input-display") == "warning")) {
        formSubmit.setAttribute("disabled","");
        return
    }

    if (!(formPrivacy1.checked) && !(formPrivacy2.checked)) {
        formSubmit.setAttribute("disabled","");
        return
    }

    formSubmit.removeAttribute("disabled");
    return;
}

function editFormMessageSiteNameWarning(val) {
    messageContainer.classList.remove("visibly-hidden")
    newInner=val.toLowerCase()
    for (var i=0; i<newInner.length; i++) {
        var letter=newInner[i];
        if (!(allowedChars.includes(letter))) {
            newInner=newInner.replaceAt(i,"-")
        }
    }
    if (hasRepeatedDashes(newInner)) { newInner=replaceRepeatedDashes(newInner) }
    messageSpan.innerHTML=val
}

function hideFormMessage() {
    messageContainer.classList.add("visibly-hidden")
    messageSpan.innerHTML=""
}

function hasRepeatedDashes(val) {
    for (var i=0;i<val.length;i++) {
        if (val[i] == "-" && val[i+1] == "-") {
            return true
        }
    }
    return false
}

function replaceRepeatedDashes(val) {
    return listToStr(replaceRepeatedDashesRecursion(val.split(""))))
}
```

```
function listToStr(lst) {
    var out=""
    for (var i=0;i<lst.length;i++) {
        out=out+lst[i]
    }
    return out
}

function replaceRepeatedDashesRecursion(val) {
    for (var i=0;i<val.length;i++) {
        if (val[i] == "-" && val[i+1] == "-") {
            val.splice(i+1,1)
            val=replaceRepeatedDashesRecursion(val)
        }
    }
    return val
}

function verifyNameField() {
    var nameInput = document.getElementById("new_site_name");
    var val = nameInput.value;

    hideFormMessage()

    if (val.length < 1) { return "inactive" }
    if (val.length < 4) { return "danger" }

    var check=true
    for (var i=0;i<val.length;i++) {
        var letter = val[i]
        if (requiredChars.includes(letter)) { check=false }
    }

    if (check) { return "danger" }

    var sitenames = ["helloworld"]

    if (flashedSiteNames.includes(val)) {
        editFormMessage("A site with this name already exists!");
        return "danger"
    }

    for (var i=0;i<val.length;i++) {
        var letter = val[i]
        if (!(allowedChars.includes(letter))) {
            editFormMessageSiteNameWarning(val);
            return "warning"
        }
    }

    if (hasRepeatedDashes(val)) {
        editFormMessageSiteNameWarning(replaceRepeatedDashes(val));
    }
}
```

```

        return "warning"
    }

    hideFormMessage()
    return "success"
}

var requiredChars = "qwertyuiopasdfghjklzxcvbnm1234567890"
var allowedChars = "qwertyuiopasdfghjklzxcvbnm-_1234567890";
var formSubmit = document.getElementById("new_site_form_submit");
var formName = document.getElementById("new_site_name");
var formDesc = document.getElementById("new_site_desc");
var formPrivacy1 = document.getElementById("new_site_privacy_visible");
var formPrivacy2 = document.getElementById("new_site_privacy_hidden");

var messageContainer = document.querySelector(
    ".new-site-form .form-input-container.one .message-container");

var messageSpan = document.querySelector(
    ".new-site-form .form-input-container.one .message-container
    .message-container-jsedit");

formName.addEventListener("keyup", (event) => {
    formName.setAttribute("data-form-input-display", verifyNameField())
    checkFormSubmitButton();
})

formPrivacy1.addEventListener("click", checkFormSubmitButton)
formPrivacy2.addEventListener("click", checkFormSubmitButton)

formSubmit.addEventListener("click", (event)=>{
    if (!(formSubmit.getAttribute("disabled"))){
        formSubmit.children[0].innerHTML = "CREATING SITE..."
    }
})

```

/static/js/site-edit.js

```

// source https://stackoverflow.com/questions/14766951/
// transform-numbers-to-words-in-lakh-crore-system Juan Gaitan

var num = "zero one two three four five six seven eight nine ten eleven twelve
thirteen fourteen fifteen sixteen seventeen eighteen nineteen".split(" ");
var tens = "twenty thirty forty fifty sixty seventy eighty ninety".split(" ");

function number2words(n){
    if (n < 20) return num[n];
    var digit = n%10;

    if (n < 100) {
        return tens[~~(n/10)-2] + (digit? "-" + num[digit]: "");
    }
}

```

```

}

if (n < 1000) {
    return num[~~(n/100)]+" hundred"+(n%100 == 0? ":" and "
+ number2words(n%100));
}
return number2words(~~(n/1000))+ " thousand"+(n%1000 != 0? " "
+ number2words(n%100):"");
}

// source https://www.codegrepper.com/profile/code-grepper

function capitalizeWords(string) {
    return string.replace(/(?:^|\s)\S/g,
        function(a) { return a.toUpperCase() })
}

// endsource

function addElement(parent,name,src,css,js,type,order) {
    // (int) parent refers to the id of the element that contains this one.
    // If set to none, the code assumes it is in the top level
    // (ie it is a section div).

    // (str) name is used to refer to this element in the gui.
    // It does not have to be unique

    // (str) src refers to the html code of this element,
    // in the form of a url starting in /static/data/

    // (str) css refers to the css code of this element,
    // in the form of a url starting in /static/data/

    // (str) js refers to the js code of this element,
    // in the form of a url starting in /static/data/

    // (str) type refers to the class of element that this is
    // (eg section, headline, table, quote, image)

    // (int) order is an integer defining how far down in the parent
    // element this element is. Used in conjunction with the other children
    // elements in the parent. If two have the same order, it will then refer to
    // their element id. 0 will mean it is at the top.

    var toplevel=false;
    if (parent == null) { toplevel=true }
    else { if (parent >= siteDat.length) {
        console.log("Parent ID larger than list of elements");return false; } }

    var id=0;
    if (siteDat.length > 0) { siteDat[siteDat.length-1]["id"]+1; }

    siteDat.add(

```

```

    {
      "name":name,
      "locked":false,
      "src":src,
      "css":css,
      "js":js,
      "type":type,
      "parent":parent,
      "id":id,
      "toplevel":toplevel,
      "order":order,
    }
  )
}

function setSectionNavbar(text) {
  text=text.split(/[\r\n]+/g);out="";
  for (var i=0; i<text.length; i++) {
    out=out+sectionSelectorNavItem
    .replace("[i]",number2words(i+1).replace(" ",""))
    .replace("[n1]",text[i].replace(" ",""))
    .replace("[n2]",capitalizeWords(text[i])))
  }
  sectionSelectorNav.querySelector("ul.section-selector-nav-list").innerHTML=out;
}

function sectionNavbarSetSelected() {
  sectionSelectorNav.querySelector(
    `ul.section-selector-nav-list li a.link`)
    .style.opacity=0.75;

  sectionSelectorNav.querySelector(
    `ul.section-selector-nav-list li.${sectionSelectorNavSelected} a.link`)
    .style.opacity=1;

  sectionSelectorDisplayPreview();
}

function sectionSelectorDisplayPreview() {
  function layer1(text) {
    path=`../../../../static/html/sections/
${text.split(/[\r\n]+/g)[sectionSelectorNavSelectedInt-1]}`;

    fetch(path+"/css.css")
      .then( response => {
        if (!response.ok) { throw new Error(`HTTP error: ${response.status}`) }
        return response.text();})
      .then( text0=>sectionSelectorList.innerHTML=sectionSelectorList.innerHTML
        +`<style>${text0}</style>`)

    fetch(path+"/files")
      .then( response => {

```

```

        if (!response.ok) { throw new Error(`HTTP error: ${response.status}`) }
        return response.text();
    .then( text1 => layer2(path, text1) )
}

function layer2(path, text) {
    text = text.split(/[\r\n]+/g).filter(
        function(value, index, arr){ return value != "" });
    for (var i=0; i<text.length; i++) {
        layer3(`#${path}/${text[i]}`)
        //fetch(`#${path}/${text[i]}`)
        // .then( response => {
        //     if (!response.ok){throw new Error(`HTTP error: ${response.status}`)}
        //     return response.text();
        // .then( text1 => layer3(text1) )
    }
}

function layer3(text) {
    fetch(text)
    .then( response => {
        if (!response.ok) { throw new Error(`HTTP error: ${response.status}`) }
        return response.text();
    .then( text2 => layer4(text2) )
}
}

function layer4(text) {
    sectionSelectorList.innerHTML = sectionSelectorList.innerHTML + text

    previewSections = sectionSelectorList.querySelectorAll("[data-preview]")
    previewSections.forEach((e)=>{
        e.style.cursor = "pointer"
        e.style.marginBottom = "32px"
    })
}

sectionSelectorList.innerHTML = "";

fetch("../static/html/sections/classes")
    .then( response => {
        if (!response.ok) { throw new Error(`HTTP error: ${response.status}`) }
        return response.text();
    .then( text => layer1(text) )
}

function parseXml(xml) {
    var parser = new DOMParser();
    return parser.parseFromString(xml,"text/xml");
}

function parseXmlFile(path) {
    fetch(path)
    .then( response => {

```

```
    if (!response.ok) { throw new Error(`HTTP error: ${response.status}`) }
    return response.text()
  .then( xml => {xml = parseXml(xml);return xml} )
}

function insertWebsite(html) { builder.innerHTML=html; }

var builder = document.getElementById("contains_site");
var addSectionBtn = document.getElementById("localnav_add_section_btn");
var sectionSelectorNavSelected="one";
var sectionSelectorNavSelectedInt=1;
var sectionSelectorContainer=document.querySelector(
  ".application-content .section-selector-container");
var sectionSelectorList=document.getElementById("section_selector_list");
var sectionSelectorNav=document.getElementById("section_selector_nav");

var sectionSelectorNavItem=`<li class="section-selector-nav-item [i]">
<a class="link unformatted" id="section_sele ctor_nav_[n1]"><
span class="text bold">[n2]</span></a></li>`;

var sectionClassList="";
var previewSections;

addSectionBtn.addEventListener("click",() => {
  if (!(sectionSelectorContainer.classList.contains("shown"))) {
    sectionSelectorContainer.classList.add("shown");
    document.querySelector(".lightbox-mask").classList.add("shown")
    sectionSelectorNavSelected="one";
    sectionNavbarSetSelected()
  }
});

document.querySelector(".application-content .section-selector-exit-btn")
.addEventListener("click",() => {

  document.querySelector(".application-content .section-selector-container")
  .classList.remove("shown")

  document.querySelector(".lightbox-mask")
  .classList.remove("shown")

});

document.querySelector(".lightbox-mask").addEventListener("click",() => {
  console.log(1);

  document.querySelector(".application-content .section-selector-container")
  .classList.remove("shown")

  document.querySelector(".lightbox-mask")
  .classList.remove("shown")
});
```

```

// section selector navbar content

fetch("../static/html/sections/classes")
  .then( response => {
    if (!response.ok) { throw new Error(`HTTP error: ${response.status}`) }
    return response.text();
  })
  .then( text => setSectionNavbar(text) )

parseXmlFile("../static/data/userData/test/sites/testing-site/site.xml");

fetch("../static/data/userData/test/sites/testing-site/files/1.html")
  .then( response => {
    if (!response.ok) { throw new Error(`HTTP error: ${response.status}`) }
    return response.text();
  })
  .then( text => insertWebsite(text))

```

/templates/

/templates/base.html

```

<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-type" content="text/html; charset=utf-8">
    <meta http-equiv="Content-type" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Kraken</title>

    <!-- Site Meta -->
    <meta name="title" content="Kraken">
    <meta name="description" content="SiteDescription">
    <meta name="robots" content="index, follow">
    <meta name="language" content="English">
    <meta name="author" content="Tom_gxz">
    <meta name="keywords" content="keywords">

    <!-- Site Icons -->
    <!-- navbarLogoColor is a jinja variable that is defined in files that
    extend from this one. It defines what colour the logo should be -
    primary, secondary, or gradient -->

    <link rel="apple-touch-icon" sizes="512x512" href="{{url_for('static',
      filename='img/icon/512-512/kraken-icon-png-' + navbarLogoColor + '-128-128.png'
    )}}>

    <link rel="icon" type="image/png" sizes="128x128" href="{{url_for('static',
      filename='img/icon/128-128/kraken-icon-png-' + navbarLogoColor + '-128-128.png'
    )}}>

```

```
)}}}">

<link rel="mask-icon" href="{{url_for('static',
filename='img/icon/mask-icon/mask-icon.svg')}}">

<meta name="apple-mobile-web-app-title" content="Kraken">
<meta name="application-name" content="SiteName">
<meta name="theme-color" content="#SiteColor">

<link rel="canonical" href="CanonicalUrl">

<!-- Open Graph Protocol -->
<meta property="og:site_name" content="Kraken">
<meta property="og:title" content="Kraken">
<meta property="og:description" content="SiteDescription">
<meta property="og:image:type" content="image/png">
<meta property="og:image:width" content="512">
<meta property="og:image:height" content="512">
<meta property="og:image" content="{{url_for('static',
filename='img/icon/tab-icon/tab-icon-512-512.png')}}">
<meta property="og:type" content="website">
<meta property="og:url" content="PageUrl">

<!-- Twitter Embed -->
<meta property="twitter:card" content="summary">
<meta property="twitter:site" content="@TwitterHandle">
<meta property="twitter:title" content="Kraken">
<meta property="twitter:description" content="SiteDescription">
<meta property="twitter:image" content="{{url_for('static',
filename='img/icon/tab-icon/tab-icon-512-512.png')}}">
<meta property="twitter:url" content="PageUrl">

<!-- Font Awesome Imports -->
<script src="https://kit.fontawesome.com/73a2cc1270.js"></script>
<link rel="stylesheet"
href="https://pro.fontawesome.com/releases/v6.0.0-beta3/css/all.css">

<!-- Internal Stylesheet Imports -->
<link href="{{url_for('static', filename='css/main.css')}}" rel="stylesheet" type="text/css" />

<link href="{{url_for('static', filename='css/build.css')}}" rel="stylesheet" type="text/css" />

</head>
<body>

<div class="page">
<div class="application-container">

    <!-- Navigation bar, docked on the left hand side -->
    <!-- Contains the logo as a link to the homepage at the top, and a
```

```
hamburger at the bottom -->

<nav class="globalnav globalnav-vertical">
  <div class="globalnav-content">
    <div class="globalnav-list">
      <div class="globalnav-logo">
        <a class="globalnav-link globalnav-link-home link unformatted"
           href="{{ url_for('main_home') }}>
          
          <span class="globalnav-link-hidden-text visibly-hidden">
            Kraken
          </span>
        </a>
      </div>
      <ul class="globalnav-list">
        <li class="globalnav-item one fake" role="button"></li>
        <li class="globalnav-item two" role="button">
          <div class="hamburger hamburger--collapse js-hamburger"
               id="globalnav-hamburger">
            <div class="hamburger-box">
              <div class="hamburger-inner"></div>
            </div>
          </div>
        </li>
      </ul>
    </div>
  </div>
</nav>

<!-- Floating option modal for the navbar, which is opened and closed
via the hamburger in the navigation bar --&gt;

&lt;div class="globalnav-floating-options"&gt;
  <!-- URL links are left blank for now as the pages have not yet
  been created --&gt;

  &lt;a class="globalnav-floating-option one" href=""&gt;
    &lt;span class="globalnav-floating-option-content text header small"&gt;
      My Sites&lt;/span&gt;
  &lt;/a&gt;

  &lt;a class="globalnav-floating-option two" href=""&gt;
    &lt;span class="globalnav-floating-option-content text header small"&gt;
      Settings&lt;/span&gt;
  &lt;/a&gt;

  &lt;a class="globalnav-floating-option three" href=""&gt;
    &lt;span class="globalnav-floating-option-content text header small"&gt;
      Logout&lt;/span&gt;
  &lt;/a&gt;
&lt;/div&gt;</pre>
```

```

<!-- Backdrop behind nav bar modal to apply a darkness filter behind
the modal -->

<div class="globalnav-floating-options-backdrop"></div>

<!-- External Script Imports -->
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.0/
jquery.min.js"></script>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"
crossorigin="anonymous"></script>

<!-- Internal Script Imports -->
<script src="{{url_for('static', filename='js/main.js')}}"></script>
<script src="{{url_for('static', filename='js/globalnav-floating-options.
js')}}"></script>

{% block content %}
{% endblock %}

</div>
</div>

</body>
</html>

```

/templates/home-nosite.html

```

{% extends "base.html" %}

{% set navbarLogoColor = "primary" %}
{% set navbarOptionsEnabled = True %}

{% block content %}

<link href="{{url_for('static', filename='css/home.css')}}" rel="stylesheet"
type="text/css" />
<div class="application-content">
<div class="text-header-container">
    <h2 class="text header large dark one">Welcome, {{current_user.name}}</h2>
</div>
<div class="empty-container">
    <div class="empty-image"></div>
    <div class="empty-text-containner">
        <h4 class="text header dark one">Looks Pretty Empty Here...</h4>
        <a class="text header two link primary" href="{{url_for('site_create')}}">
            Maybe you should create a new site?</a>
        </div>
    </div>
</div>
</div>

```

```
{% endblock %}
```

/templates/home-sites.html

```
{% extends "base.html" %}

{% set navbarLogoColor = "primary" %}
{% set navbarOptionsEnabled = True %}

{% set sites = get_flashed_messages()[0] %}

{% block content %}

<link href="{{url_for('static', filename='css/home.css')}}" rel="stylesheet"
type="text/css" />
<div class="application-content">

<div class="text-header-container">
    <h2 class="text header large dark one">Welcome, {{current_user.name}}</h2>
</div>

<div class="site-divs-container" style="display:flex;flex-wrap:wrap">

    {% for site in sites %}

        <a class="site-div link notformatted" href="{{url_for('site_edit_home',
name=site[0],site=site[1])}}" style="background-color:
            {%if site[3]%}{{site[3]}}+bb'{%else%}var(--colors-primary){%endif%}">

            {% if site[2] %}
                <div class="site-div-private-watermark" style="
                    position:absolute;opacity:0.5;font-size:136px;right:16px;top:16px;">
                    <i class="faicon fa-regular fa-lock"></i>
                </div>
            {% else %}
                <div class="site-div-public-watermark" style="
                    position:absolute;opacity:0.5;font-size:136px;right:16px;top:16px;">
                    <i class="faicon fa-regular fa-book-bookmark"></i>
                </div>
            {% endif %}

            <h5 class="site-div-title text large one">{{ site[1] }}</h5>
        </a>

    {% endfor %}

    <a class="site-div link notformatted" style="
        background-color:var(--colors-primary-light);display:flex;
        align-items:center;justify-content:center" href="{{url_for('site_create')}}">
        <h5 class="site-div-title text header jumbo small one center">
```

```

        Create New Site
    </h5>
</a>

</div>

</div>

{% endblock %}

```

/templates/login.html

```

{% extends "base.html" %}

{% set navbarLogoColor = "secondary" %}
{% set navbarOptionsEnabled = False %}
{% set messages = get_flashed_messages()[0] %}

{% block content %}

<link href="{{url_for('static', filename='css/auth.css')}}" rel="stylesheet"
type="text/css" />

<div class="application-content">
    <div class="text-header-container">
        <h2 class="text header xl dark one">Kraken - Login</h2>
        <ul class="header-options">
            <li class="header-option header-option-login active notextselect">
                <h4 class="text header bold">Login</h4>
            </li>
            <li class="header-option header-option-signup notextselect"
                onclick="window.location.href=`{{ url_for('auth_signup') }}`">
                <h4 class="text header bold">Signup</h4>
            </li>
        </ul>
    </div>
    <div class="field-container active">
        <span class="field-warning text italic">
            {% if messages[0] %}
                {{ messages[1] }}
            {% endif %}
        </span>
    <form class="field-options" method="post" action="/login/">
        {% set formItems = [
            ["Username", "Username", "text", "username", false, messages[2]],
            ["Password", "Password", "password", "password", true, ""]
        ] %}

```

```

{% for item in formItems %}
<div class="field-option field-option-name">
  <h4 class="text italic">{{item[0]}}</h4>
  <div class="field-input-container">
    <input class="field-input" placeholder="{{item[1]}}"
      type="{{item[2]}}" name="{{item[3]}}" value="{{item[5]}}>
    {% if item[4] %}
      <span class="eye-reveal">
        <i class="fa-solid fa-eye"></i>
      </span>
    {% else %}
      <span class="eye-spacer"></span>
    {% endif %}
  </div>
</div>
{% endfor %}

<div class="field-option field-option-remember">
  <h4 class="text italic">Remember Me</h4>
  <div class="field-input-container">
    <input class="field-input" type="checkbox" name="remember"
      value={{messages[3]}}>
    <span class="eye-spacer"></span>
  </div>
</div>

<button class="field-submit btn secondary rounded slide" type="submit">
  <span class="btn-content text uppercase secondary">Submit</span>
</button>

</form>

</div>
</div>

<script src="..../static/js/auth.js"></script>
<script src="..../static/js/login.js"></script>

{% endblock %}

```

/templates/settings-admin.html

```

{% extends "settings-base.html" %}

{% set navbarLogoColor = "primary" %}
{% set navbarOptionsEnabled = True %}

{% set settingsSidebarActivated = 2 %}

{% block settings_content %}

```

```
<h3 class="settings-content-header text header dark">Account</h3>

<form class="settings-content-options">

  <div class="settings-content-option one">
    <h4 class="settings-content-option-title text dark bold">
      Change Username
    </h4>

    <p class="settings-content-option-caption text small danger dark">
      <span class="text small danger bold">WARNING</span>.
      Changing your username can
      <a class="text small link danger bold">create issues</a>.
    </p>

    <div class="btn primary thin rounded slide m-xs-t">
      <span class="btn-content text uppercase primary">Dew it</span>
    </div>
  </div>

  <div class="settings-content-option one">
    <h4 class="settings-content-option-title text dark bold">Change Email</h4>
    <input class="settings-content-option-input" type="text"
      name="settings_admin_email" placeholder="New Email">
    <br>

    <input class="settings-content-option-input m-xs-t" type="password"
      name="settings_admin_email_password" placeholder="Password">

    <p class="settings-content-option-caption text small dark">
      Your old and new email addresses will be sent confirmation codes
      in order to change them.
    </p>

    <div class="btn primary thin rounded slide m-xs-t">
      <span class="btn-content text uppercase primary">Dew it</span>
    </div>
  </div>

  <div class="settings-content-separator one"></div>

  <div class="settings-content-option three">
    <h4 class="settings-content-option-title text dark bold">
      Export Account Data
    </h4>

    <p class="settings-content-option-caption text small dark">
      Export all metadata, websites, and other stored information for your
      account. They will be available to download here.
    </p>
  </div>
</form>
```

```
<div class="btn primary thin rounded slide m-xs-t">
  <span class="btn-content text uppercase primary">Export Metadata</span>
</div>

<div class="btn primary thin rounded slide m-xs-t">
  <span class="btn-content text uppercase primary">Export Websites</span>
</div>

<div class="btn primary thin rounded slide m-xs-t">
  <span class="btn-content text uppercase primary">Download All</span>
</div>

</div>

<div class="settings-content-separator two"></div>

<div class="settings-content-option four">
  <h4 class="settings-content-option-title text dark bold danger">
    Archive Account
  </h4>

  <p class="settings-content-option-caption text small dark">
    This will disable your account until you wish to unlock it.
  </p>

  <div class="btn danger thin rounded slide m-xs-t">
    <span class="btn-content text uppercase danger">
      Archive your account
    </span>
  </div>
</div>

<div class="settings-content-option five">
  <h4 class="settings-content-option-title text dark bold danger">
    Reset Account
  </h4>

  <p class="settings-content-option-caption text small dark">
    This will remove all of your websites, custom code, and non-essential
    settings. The only remaining settings will be your username, name,
    email, and password. It is recommended that you export your account
    data before doing this.
  </p>

  <div class="btn danger thin rounded slide m-xs-t">
    <span class="btn-content text uppercase danger">Reset your account</span>
  </div>
</div>

<div class="settings-content-option six">
  <h4 class="settings-content-option-title text dark bold danger">
    Delete Account
  </h4>
```

```

</h4>

<p class="settings-content-option-caption text small dark">
    This will remove all trace of your account from our servers, and is an
    irreversible action. It is recommended that you export your account data
    before doing this.
</p>

<div class="btn danger thin rounded slide m-xs-t">
    <span class="btn-content text uppercase danger">Delete your account</span>
</div>

</div>

</form>

{% endblock %}

```

/templates/settings-base.html

```

{% extends "base.html" %}

{% set navbarLogoColor = "primary" %}
{% set navbarOptionsEnabled = True %}

{% set settingsSidebarActivated = get_flashed_messages()[0] %}

{% block content %}

<link href="{{url_for('static', filename='css/settings.css')}}" rel="stylesheet"
type="text/css" />

<div class="application-content">
    <div class="text-header-container">
        <h2 class="text header large dark one">Settings</h2>
    </div>

    <div class="settings-container">
        <div class="settings-sidebar">

            <a class="settings-sidebar-item one
            {%if settingsSidebarActivated==1%} is-active{%endif%} link notformatted"
            {% if not settingsSidebarActivated==1 %}>
                <span class="settings-sidebar-item-icon fa-regular fa-user"></span>
                <span class="settings-sidebar-item-title text large">
                    Public Profile
                </span>
            </a>
        
```

```
<a class="settings-sidebar-item two"
{%if settingsSidebarActivated==2%} is-active{%endif%} link notformatted"
{% if not settingsSidebarActivated==2 %}
href="{{ url_for('settings_admin') }}"{% endif %}>

    <i class="settings-sidebar-item-icon fa-regular fa-gear"></i>
    <span class="settings-sidebar-item-title text large">
        Account
    </span>
</a>

<a class="settings-sidebar-item three"
{%if settingsSidebarActivated==3%} is-active{%endif%} link notformatted"
{% if not settingsSidebarActivated==3 %}
href="{{ url_for('settings_looks') }}"{% endif %}>

    <i class="settings-sidebar-item-icon fa-regular fa-paintbrush"></i>
    <span class="settings-sidebar-item-title text large">
        Appearance & Accessibility
    </span>
</a>

<div class="settings-sidebar-separator text one">
    Code and websites
</div>

<a class="settings-sidebar-item four"
{%if settingsSidebarActivated==4%} is-active{%endif%} link notformatted"
{% if not settingsSidebarActivated==4 %}
href="{{ url_for('settings_sites') }}"{% endif %}>

    <i class="settings-sidebar-item-icon fa-regular fa-browser"></i>
    <span class="settings-sidebar-item-title text large">
        My Websites
    </span>
</a>

<a class="settings-sidebar-item five"
{%if settingsSidebarActivated==5%} is-active{%endif%} link notformatted"
{%if not settingsSidebarActivated==5%}
href="{{ url_for('settings_code') }}"{% endif %}>

    <i class="settings-sidebar-item-icon fa-regular fa-list-timeline"></i>
    <span class="settings-sidebar-item-title text large">
        Custom Code & Elements
    </span>
</a>

<div class="settings-sidebar-separator text two"> </div>

<a class="settings-sidebar-item six"
{%if settingsSidebarActivated==6%} is-active{%endif%} link notformatted"
{% if not settingsSidebarActivated==6 %}>
```

```

    href="{{ url_for('main_help') }}"}{% endif %}>

    <i class="settings-sidebar-item-icon fa-regular fa-book-blank"></i>
    <span class="settings-sidebar-item-title text large">
        Help & Documentation
    </span>
</a>

<a class="settings-sidebar-item seven
{%if settingsSidebarActivated==7%} is-active{%endif%} link notformatted"
{%if not settingsSidebarActivated==7%}
href="{{ url_for('settings_dev') }}"}{% endif %}>

    <i class="settings-sidebar-item-icon fa-regular fa-code-simple"></i>
    <span class="settings-sidebar-item-title text large">
        Developer settings
    </span>
</a>

</div>

<div class="settings-content">

    {% block settings_content %}
    {% endblock %}

</div>
</div>
</div>

{% endblock %}

```

/templates/settings-code.html

```

{% extends "settings-base.html" %}

{% set navbarLogoColor = "primary" %}
{% set navbarOptionsEnabled = True %}

{% set settingsSidebarActivated = 5 %}

{% block settings_content %}

ToDo:

My custom code
    Displays name, type of code, size of code, and settings for it

My custom elements
    Displays name, description, and settings for it

```

```
{% endblock %}
```

/templates/settings-dev.html

```
{% extends "settings-base.html" %}

{% set navbarLogoColor = "primary" %}
{% set navbarOptionsEnabled = True %}

{% set settingsSidebarActivated = 7 %}

{% block settings_content %}

To do:

warning at the start

{% endblock %}
```

/templates/settings-looks.html

```
{% extends "settings-base.html" %}

{% set navbarLogoColor = "primary" %}
{% set navbarOptionsEnabled = True %}

{% set settingsSidebarActivated = 3 %}

{% block settings_content %}

<h3 class="settings-content-header text header dark">
    Appearance and Accessibility
</h3>

<form class="settings-content-options">

    <div class="settings-content-option one">
        <h4 class="settings-content-option-title text dark bold">
            Tab Preference
        </h4>

        <select class="settings-content-option-input"
               name="settings_looks_tab_preference">
            <option value="1">1</option>
            <option value="2">2</option>
            <option value="3">3</option>
            <option value="4" selected="selected">4 (Default)</option>
        </select>
    </div>
</form>
```

```

        <option value="5">5</option>
        <option value="6">6</option>
        <option value="8">8</option>
        <option value="10">10</option>
        <option value="12">12</option>
    </select>

    <p class="settings-content-option-caption text small dark">
        When editing and rendering code, this determines how many spaces
        represent one tab. (Doesn't do anything yet)
    </p>

</div>

</form>

{% endblock %}

```

/templates/settings-profile.html

```

{% extends "settings-base.html" %}

{% set navbarLogoColor = "primary" %}
{% set navbarOptionsEnabled = True %}

{% set settingsSidebarActivated = 1 %}
{% set avatarImagePath = get_flashed_messages()[1] %}

{% block settings_content %}

<h3 class="settings-content-header text header dark">Profile</h3>

<form class="settings-content-options">

    <div class="settings-content-option one">
        <h4 class="settings-content-option-title text dark bold">Name</h4>
        <input class="settings-content-option-input" type="text"
            name="settings_profile_name" placeholder="{{current_user.name}}">
        <p class="settings-content-option-caption text small dark">
            Your name probably isn't used much yet, but may appear in reference to
            websites that you have created. Your current name is set to
            "{{current_user.name}}".
        </p>
    </div>

    <div class="settings-content-option two">
        <h4 class="settings-content-option-title text dark bold">Profile Picture</h4>
        <div class="settings-content-option-image-upload">
            <figure class="settings-content-option-image-upload-figure"
                style="background-image:url('{{ url_for('static',
                    filename='data/userIcons/'+current_user.user_id+'.png') }})"></figure>

```

```

        </div>

        <p class="settings-content-option-caption text small dark">
            The image must be a minimum of 200x200 pixels, and in a 1:1 ratio.
            This is not operational yet, and probably won't be for a while.
        </p>
    </div>

    <div class="settings-content-separator one"></div>

    <div class="settings-content-option three">
        <h4 class="settings-content-option-title text dark bold">Bio</h4>
        <textarea class="settings-content-option-input" type="text"
            name="settings_profile_bio" maxlength=240
            placeholder="Tell us about yourself"></textarea>
    </div>

    <div class="settings-content-option four">
        <h4 class="settings-content-option-title text dark bold">Url</h4>
        <input class="settings-content-option-input" type="text"
            name="settings_profile_url">
    </div>

    <div class="settings-content-separator two"></div>

    <div class="settings-content-option settings-content-update">

        <p class="settings-content-option-caption text small dark">
            All of the above fields are optional and can be left blank.
            By filling them out, you agree that this information can be displayed
            publicly and stored in our servers. We don't have a privacy statement,
            but we probably should.
        </p>

        <button class="field-submit btn primary thin rounded slide" type="submit">
            <span class="btn-content text uppercase primary">Update Profile</span>
        </button>

    </div>

</form>

{% endblock %}

```

/templates/settings-sites.html

```

{% extends "settings-base.html" %}

{% set navbarLogoColor = "primary" %}
{% set navbarOptionsEnabled = True %}

```

```

{% set settingsSidebarActivated = 4 %}
{% set flashedSiteNames = get_flashed_messages()[1] %}

{% block settings_content %}

<h3 class="settings-content-header text header dark">My Websites</h3>

<form class="settings-content-options">

    <div class="settings-content-table">
        <div class="settings-content-table-header">
            <div class="settings-content-table-row">
                <span class="text header dark">Websites</span>
            </div>
            <div class="settings-content-table-row">
                <span class="text header small dark">@{{current_user.user_id}}</span>
            </div>
        </div>

        <div class="settings-content-table-content">

            {% for site in flashedSiteNames %}

                <div class="settings-content-table-row"
                    {% if site==flashedSiteNames[-1] %}last-row{% endif %}>
                    <span class="settings-content-table-icon text dark">
                        <i class="fa-regular {% if site[2] %}fa-lock{% else %}fa-book-bookmark{%endif%}"></i>
                    </span>

                    <span class="settings-content-table-row-title text dark">
                        <a href='{{url_for("site_edit_home",name=site[0],site=site[1])}}'
                            class="text link dark notformatted">@{{site[0]}}/{{site[1]}}</a>
                    </span>

                    <span class="settings-content-table-row-size text dark">
                        {{site[3]}}
                    </span>

                    <span class="settings-content-table-settings text dark">
                        <a href='{{url_for("site_edit_home",name=site[0],site=site[1])}}'
                            class="text link primary notformatted">Website Settings</a>
                    </span>
                </div>

            {% endfor %}

        </div>
    </div>

</form>

```

```
{% endblock %}
```

/templates/signup.html

```
{% extends "base.html" %}

{% set navbarLogoColor = "secondary" %}
{% set navbarOptionsEnabled = False %}
{% set messages = get_flashed_messages()[0] %}

{% block content %}

<link href="{{url_for('static', filename='css/auth.css')}}" rel="stylesheet"
type="text/css" />
<div class="application-content">
<div class="text-header-container">
<h2 class="text header xl dark one">Kraken - Signup</h2>
<ul class="header-options">
<div class="header-option header-option-login notextselect"
onclick="window.location.href=`{{ url_for('auth_login') }}`">
<h4 class="text header bold">Login</h4>
</div>
<div class="header-option header-option-signup active notextselect">
<h4 class="text header bold">Signup</h4>
</div>
</ul>
</div>
<div class="field-container active">
<span class="field-warning text italic">
{% if messages[0] %}
{{ messages[1] }}
{% endif %}
</span>

<form class="field-options" method="post" action="/signup/">

{% set formItems = [
["Name", "Name", "text", "name", false, messages[2]],
["Email", "name@domain.com", "email", "email", false, messages[3]],
["Username", "Username", "text", "username", false, messages[4]],
["Password", "Password", "password", "password", true, ""],
["Repeat Password", "Again :/", "password", "password", "password-repeat", ""]
]
%}

{% for item in formItems %}
<div class="field-option field-option-name">
<h4 class="text italic">{{item[0]}}</h4>
<div class="field-input-container">
<input class="field-input" placeholder="{{item[1]}}"
```

```

        type="{{item[2]}}" name="{{item[3]}}" value="{{item[5]}}">
        {% if item[4] %}
        <span class="eye-reveal">
            <i class="fa-solid fa-eye"></i>
        </span>
        {% else %}
        <span class="eye-spacer"></span>
        {% endif %}
    </div>
</div>
{% endfor %}

<button class="field-submit btn secondary rounded slide" type="submit">
    <span class="btn-content text uppercase secondary">Submit</span>
</button>

</form>

</div>
</div>

<script src="../static/js/auth.js"></script>
<script src="../static/js/signup.js"></script>

{% endblock %}

```

/templates/site-create-base.html

```

{% extends "base.html" %}

{% set navbarLogoColor = "primary" %}
{% set navbarOptionsEnabled = True %}

{% block content %}

<link href="{{url_for('static', filename='css/site-create.css')}}" rel="stylesheet" type="text/css" />

<div class="application-content">
    <div class="text-header-container">
        <h2 class="text header large dark one">Create a new site</h2>
    </div>

    <div class="main">
        <div class="main-content thin">
            {%block site_create_base%}
            {%endblock%}
        </div>
    </div>
</div>

```

```
{% endblock %}
```

/templates/site-create-options-1.html

```
{% extends "site_create_base.html" %}

{% set navbarLogoColor = "primary" %}
{% set navbarOptionsEnabled = True %}

{% block site_create_base %}

    <p class="text dark">Choose a color scheme!</p>
    <div class="horizontal-separator one m-s-v"></div>

    <form class="new-site-form two" method="post">

        <div class="light-dark-selector-container">
            <div class="light-dark-selector">
                <h2 class="text large center one">Light or dark mode?</h2>
                <div class="button-container">

                    <div class="btn primary thin rounded slide from-left m-xs-t"
                        id="new_site_lightModeToggle">
                        <span class="btn-content text uppercase primary notextselect">
                            Light
                        </span>
                    </div>

                    <div class="btn secondary thin rounded slide from-right m-xs-t"
                        id="new_site_darkModeToggle">
                        <span class="btn-content text uppercase secondary notextselect">
                            Dark
                        </span>
                    </div>
                </div>
            </div>
        </div>

        <div class="horizontal-separator one m-s-v"></div>

        <div class="horizontal-separator two m-s-v"></div>

        <div class="color-options">
            <div class="input-slider-option one">
                <span class="text dark bold">Light & Dark Bounds</span>
                <div class="input-slider-and-number">
                    <input class="input sliding-input" type="range" min="0" max="100"
                        value="100" name="new_site_colors_light_dark_bounds"
                        id="new_site_colors_light_dark_bounds_slider">
                </div>
            </div>
        </div>
    </form>
</div>
```

```
<input class="input small-number-input" type="text"
    pattern="[-+]?d*" min="-100" max="100"
    id="new_site_colors_light_dark_bounds_number">
</div>
</div>
<div class="input-slider-option two">
    <span class="text dark bold">Monochromatic Temperature</span>
    <div class="input-slider-and-number">
        <input class="input sliding-input" type="range" min="-100"
            max="100" value="0" name="new_site_colors_monochromatic_tint"
            id="new_site_colors_monochromatic_temperature_slider">
        <input class="input small-number-input" type="text"
            pattern="[-+]?d*" min="-100" max="100"
            id="new_site_colors_monochromatic_temperature_number">
    </div>
</div>
</div>

<div class="horizontal-separator three m-s-v"></div>

<div class="color-display-container">

    <div class="color-display light-dark-display">
        <div class="color-single-card light-color"
            style="background-color:#ffffff;color:#000000">
            <span class="color-code text uppercase center">#ffffff</span>
        </div>
        <div class="color-single-card dark-color"
            style="background-color:#000000;color:#ffffff">
            <span class="color-code text uppercase center">#000000</span>
        </div>
    </div>

    <div class="color-display main-color-display">

        <div class="color-triple-card primary-color">
            <input type="color" class="color-card-picker-input"
                id="new_site_colors_primary_picker" value="#e63946">

            <div class="color-triple-card-main"
                style="background-color:#e63946;color:#000000">
                <span class="color-code text uppercase center">#e63946</span>
            </div>

            <div class="color-triple-card-sub-container">
                <div class="color-triple-card-sub one"
                    style="background-color:#ba2b36;color:#000000"></div>
                <div class="color-triple-card-sub two"
                    style="background-color:#f2414f;color:#000000"></div>
            </div>
        </div>

        <div class="color-triple-card secondary-color">

```

```
<input type="color" class="color-card-picker-input"
id="new_site_colors_secondary_picker" value="#457b9d">
<div class="color-triple-card-main"
style="background-color:#457b9d;color:#000000">
    <span class="color-code text uppercase center">#457b9d</span>
</div>
<div class="color-triple-card-sub-container">
    <div class="color-triple-card-sub one"
style="background-color:#3f708f;color:#000000"></div>
    <div class="color-triple-card-sub two"
style="background-color:#508eb5;color:#000000"></div>
</div>
</div>

<div class="color-triple-card accent-color">
    <input type="color" class="color-card-picker-input"
id="new_site_colors_accent_picker" value="#a8dadc">
    <div class="color-triple-card-main"
style="background-color:#a8dadc;color:#000000">
        <span class="color-code text uppercase center">#a8dadc</span>
    </div>
    <div class="color-triple-card-sub-container">
        <div class="color-triple-card-sub one"
style="background-color:#93c5c7;color:#000000"></div>
        <div class="color-triple-card-sub two"
style="background-color:#b4ebed;color:#000000"></div>
    </div>
</div>

</div>

<div class="color-display grey-display">
    <div class="color-columns grey-colors expanding-columns-container">

        <div class="color-column expanding-column g900"
style="background-color:#303030;color:#ffffff">
            <span class="color-code expanding-text text uppercase center">
                #303030</span></div>
        <div class="color-column expanding-column g800"
style="background-color:#474747;color:#ffffff">
            <span class="color-code expanding-text text uppercase center">
                #474747</span></div>
        <div class="color-column expanding-column g700"
style="background-color:#5e5e5e;color:#ffffff">
            <span class="color-code expanding-text text uppercase center">
                #5e5e5e</span></div>
        <div class="color-column expanding-column g600"
style="background-color:#757575;color:#ffffff">
            <span class="color-code expanding-text text uppercase center">
                #757575</span></div>
        <div class="color-column expanding-column g500"
style="background-color:#8c8c8c;color:#000000">
            <span class="color-code expanding-text text uppercase center">
                #8c8c8c</span></div>
    </div>
</div>
```

```

#8c8c8c</span></div>
<div class="color-column expanding-column g400"
style="background-color:#a3a3a3;color:#000000">
<span class="color-code expanding-text text uppercase center">
#a3a3a3</span></div>
<div class="color-column expanding-column g300"
style="background-color:#bababa;color:#000000">
<span class="color-code expanding-text text uppercase center">
#bababa</span></div>
<div class="color-column expanding-column g200"
style="background-color:#d1d1d1;color:#000000">
<span class="color-code expanding-text text uppercase center">
#d1d1d1</span></div>
<div class="color-column expanding-column g100"
style="background-color:#e8e8e8;color:#000000">
<span class="color-code expanding-text text uppercase center">
#e8e8e8</span></div>

</div>
</div>

</div>

<div class="submit-container">
<button class="field-submit btn primary thin rounded slide" type="submit">
<span class="btn-content text uppercase primary">Continue</span>
</button>
</div>

<input id="color-output" class="visibly-hidden" type="text" value="."
name="new_site_color_options_dict">

</form>

<script src="{{url_for('static', filename='js/colorConversion.js')}}">
</script>
<script src="{{url_for('static', filename='js/site-create-options-1.js')}}">
</script>

{% endblock %}

```

/templates/site-create-options-2.html

```

{% extends "site_create_base.html" %}

{% set navbarLogoColor = "primary" %}
{% set navbarOptionsEnabled = True %}

{% set fontsList = [
["Lexend", "Roboto", True, True],
```

```

["Prata","Lato",True,True],
["DM Sans","Catamaran",True,True],
["Titillium Web","Raleway",True,True],
["Caudex","PT Mono",True,True],
["Noto Serif Display","Lora",True,True],
["Staatliches","Syne Mono",True,True],


]

%}

{% block site_create_base %}

<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

<p class="text dark">
    Choose font family - individual elements can be customised
</p>

<div class="horizontal-separator one m-s-v"></div>

<form class="new-site-form three" method="post">

    <div class="text-options">
        {%for fontList in fontsList%}

            {%set counter=fontsList.index(fontList)+1%}

            {%set headerFont=fontList[0]%
            {%set paraFont=fontList[1]%

                <div class="text-option {{counter}}{%if counter==1%} active{%endif%}">

                    {%if fontList[2]%
                        <link href="https://fonts.googleapis.com/css2?family={{headerFont}}&display=swap" rel="stylesheet">
                    {%endif%}

                    {%if fontList[3]%
                        <link href="https://fonts.googleapis.com/css2?family={{paraFont}}&display=swap" rel="stylesheet">
                    {%endif%}

                    <div class="text-option-header text header dark one"
                        style="font-family:'{{headerFont}}'">{{headerFont}}</div>

                    <div class="text-option-paragraph text two"
                        style="font-family:'{{paraFont}}'">Paragraph text - {{paraFont}}</div>

                    <input class="visibly-hidden text-option-list"
                        value="{{headerFont}},{{paraFont}}" name="new_site_font_face_list_&nbsp;
                    {%if counter==1%}active{%else%}inactive{%endif%}">

```

```

        </div>

        {%endfor%}
    </div>

    <div class="submit-container">
        <button class="field-submit btn primary thin rounded slide" type="submit">
            <span class="btn-content text uppercase primary">Continue</span>
        </button>
    </div>

</form>

<script src="{{url_for('static', filename='js/colorConversion.js')}}">
</script>

<script src="{{url_for('static', filename='js/site-create-options-2.js')}}">
</script>

{% endblock %}

```

/templates/site-create-options-3.html

```

{% extends "site_create_base.html" %}

{% set navbarLogoColor = "primary" %}
{% set navbarOptionsEnabled = True %}

{% block site_create_base %}

<p class="text dark">Choose default styles</p>
<div class="horizontal-separator one m-s-v"></div>

<form class="new-site-form four" method="post">

    <div class="button-options">
        <div class="button-option-container">
            <div class="button-option slide-or-static">
                <div class="btn secondary thin rounded slide option-true">
                    <span class="btn-content text uppercase">Sliding Button</span>
                </div>

                <div class="btn secondary thin rounded option-false">
                    <span class="btn-content text uppercase">Solid Button</span>
                </div>
            </div>
        </div>
    </div>

    <div class="button-option-container">
        <div class="button-option corner-options"
            style="opacity:0;visibility:hidden;">

```

```
<div class="btn secondary thin square slide option-true">
  <span class="btn-content text uppercase">Squared</span>
</div>

<div class="btn secondary thin rounded option-false">
  <span class="btn-content text uppercase">Rounded</span>
</div>

<div class="btn secondary thin pill option-false">
  <span class="btn-content text uppercase">Pill</span>
</div>
</div>

<div class="button-option-container">
  <div class="button-option thin-or-large"
    style="opacity:0;visibility:hidden;">
    <div class="btn secondary thin rounded slide option-true">
      <span class="btn-content text uppercase">Thin</span>
    </div>
  </div>
</div>

<div class="button-option-container">
  <div class="button-option left-or-right"
    style="opacity:0;visibility:hidden;">
    <div class="btn secondary thin rounded slide from-left option-true">
      <span class="btn-content text uppercase">From Left</span>
    </div>
  </div>
</div>
</div>

<div class="horizontal-separator two m-s-v"></div>

<div class="submit-container">
  <button class="field-submit btn primary thin rounded slide"
    type="submit" disabled>
    <span class="btn-content text uppercase primary">Continue</span>
  </button>
</div>

<input id="style-option-output" class="visibly-hidden" type="text"
  value=". " name="new_site_style_options_list">
```

```

</form>

<script src="{{url_for('static', filename='js/colorConversion.js')}}">
</script>
<script src="{{url_for('static', filename='js/site-create-options-3.js')}}">
</script>

{% endblock %}

```

/templates/site-create.html

```

{% extends "site_create_base.html" %}

{% set navbarLogoColor = "primary" %}
{% set navbarOptionsEnabled = True %}

{% block site_create_base %}



Want to import an exported site?
    <a class="link text primary">Import a website.</a>



</div>

<form class="new-site-form one" method="post">

    <div class="form-input-container one">
        <div class="form-input-content-column">
            <span class="text large dark one">Owner</span>
            <span class="text large dark two">
                <span>@{{current_user.user_id}}</span>
                <span class="m-m-l m-s-r">/</span>
            </span>
        </div>
        <div class="form-input-content-column">

            <span class="text large dark one">
                Website Name
                <sup class="text large danger">*</sup>
            </span>

            <input id="new_site_name" class="new-site-input text dark two input small-text-input" data-form-input-display="inactive" type="text" name="new_site_name">

        </div>

        <div class="message-container m-s-t text small one visibly-hidden">
            Your site name will look like: <span class="message-container-jsedit">
        </span>
    </div>


```

```
<p class="text dark m-s-t two">
    The name must be at least 4 characters long, and contain only
    lowercase alphanumeric characters, dashes, underscores and periods.
    Any illegal characters will be converted into dashes.
    It must also be unique! If you need inspiration for a name,
    you ain't gonna get any from me :)</p>
</div>

<div class="horizontal-separator two m-s-v"></div>

<div class="form-input-container three">
    <span class="text large dark one">Description (Optional)</span>
    <input id="new_site_desc" class="new-site-input text dark two input
    small-text-input" type="text" name="new_site_desc">
</div>

<div class="horizontal-separator three m-s-v"></div>

<div class="form-input-container two">
    <div class="input-checkbox">
        <input class="input-checkbox-input" name="new_site_privacy"
        id="new_site_privacy_visible" type="radio" value="public">
        <span class="input-checkbox-icon">
            <i class="faicon fa-regular fa-book-bookmark"></i>
        </span>
        <div class="input-checkbox-text-container">
            <span class="input-checkbox-title text bold one">Public</span>
            <span class="input-checkbox-caption text small two">
                Anyone online can see this website. Only you can edit it.
            </span>
        </div>
    </div>
</div>

<div class="input-checkbox">
    <input class="input-checkbox-input" name="new_site_privacy"
    id="new_site_privacy_hidden" type="radio" value="private">
    <span class="input-checkbox-icon"><i class="faicon fa-regular fa-lock"
    style="color: var(--colors-warning)"></i></span>
    <div class="input-checkbox-text-container">
        <span class="input-checkbox-title text bold one">Private</span>
        <span class="input-checkbox-caption text small two">
            Only people who you allow can view the website.
        </span>
    </div>
</div>

<div class="horizontal-separator three m-s-v"></div>

<button class="field-submit btn primary thin rounded slide" type="submit"
disabled=true id="new_site_form_submit">
    <span class="btn-content text uppercase primary">Create Site</span>
</button>
```

```

        </div>

    </form>

    <script>
        var flashedSiteNames = "{{get_flashed_messages()[0]}}".split(",")
    </script>

    <script src="{{url_for('static', filename='js/site-create.js')}}"></script>

    {% endblock %}

```

/templates/site-edit-home.html

```

{% extends "base.html" %}

{% set navbarLogoColor = "gradient" %}
{% set navbarOptionsEnabled = True %}

{% set settingsSidebarActivated = get_flashed_messages()[0] %}
{% set currentName = get_flashed_messages()[1] %}
{% set currentSite = get_flashed_messages()[2] %}

{% block content %}

<link href="{{url_for('static', filename='css/settings.css')}}" rel="stylesheet" type="text/css" />

<div class="application-content">
    <div class="text-header-container">
        <h2 class="text header large dark one">{{currentSite}}</h2>
    </div>

    <div class="settings-container">
        <div class="settings-sidebar">

            <a class="settings-sidebar-item one"
                {%if settingsSidebarActivated==1%} is-active{%endif%} link notformatted"
                {% if not settingsSidebarActivated==1 %}
                href="{{ url_for('site_edit_home',name=currentName,site=currentSite) }}"
                {% endif %}>

                <i class="settings-sidebar-item-icon fa-regular fa-browser"></i>
                <span class="settings-sidebar-item-title text large">Home</span>
            </a>

            <a class="settings-sidebar-item two link notformatted"
                href="{{ url_for('site_edit_app',name=currentName,site=currentSite) }}"
                <i class="settings-sidebar-item-icon fa-regular fa-pen-to-square"></i>
                <span class="settings-sidebar-item-title text large">Edit Site</span>
            </a>

        </div>
    </div>
</div>

```

```
</div>

<div class="settings-content">

    {% block settings_content %}

        {% endblock %}

    </div>
</div>

<% endblock %>
```

/templates/site-edit.html

```
{% extends "base.html" %}

{% set navbarLogoColor = "gradient" %}
{% set navbarOptionsEnabled = True %}

{% set settingsSidebarActivated = get_flashed_messages()[0] %}
{% set currentName = get_flashed_messages()[1] %}
{% set currentSite = get_flashed_messages()[2] %}

{% block content %}

<link href="{{url_for('static', filename='css/site-edit.css')}}" rel="stylesheet" type="text/css" />

<script>
  fetch('{{url_for('static', filename='data/userData/'+currentName+'/sites/' +currentSite+'/siteDat.json')}}").then(response => {
    return response.json();
  })
  .then(jsondata => console.log(jsondata));
</script>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js">
</script>

<div class="lightbox-mask"></div>

<div class="application-content">

<div class="localnav one localnav-vertical localnav-collapsible">
  <div class="localnav-content">
    <ul class="localnav-list">
      <ul class="localnav-list" style="height:30%">
        <li class="localnav-item one">
          <a class="link unformatted">
```

```
<div class="localnav-item-collapsible-icon">
    <i class="faicon fa-solid fa-layer-group"></i>
</div>
<div class="localnav-item-collapsible-text text bold primary">
    Website Pages
</div>
</a>
</li>

<li class="localnav-item two">
    <a class="link unformatted" id="localnav_add_section_btn">
        <div class="localnav-item-collapsible-icon">
            <i class="faicon fa-solid fa-circle-plus"></i>
        </div>
        <div class="localnav-item-collapsible-text text bold primary">
            Add Section
        </div>
    </a>
</li>

<li class="localnav-item three">
    <a class="link unformatted">
        <div class="localnav-item-collapsible-icon">
            <i class="faicon fa-solid fa-paintbrush"></i>
        </div>
        <div class="localnav-item-collapsible-text text bold primary">
            Website Styles
        </div>
    </a>
</li>
</ul>

<li class="localnav-item four">
    <a class="link unformatted">
        <div class="localnav-item-collapsible-icon">
            <i class="faicon fa-solid fa-gear"></i>
        </div>
        <div class="localnav-item-collapsible-text text bold primary">
            Website Settings
        </div>
    </a>
</li>

</ul>
</div>

</div>

<div class="section-selector-container">
    <div class="section-selector">
        <div class="section-selector-exit-btn">
            <i class="faicon fa-solid fa-xmark"></i>
        </div>
    </div>
</div>
```

```
</div>

<div class="section-selector-sidebar">
    <div class="section-selector-nav" id="section_selector_nav">
        <div class="section-selector-nav-content">
            <ul class="section-selector-nav-list">
                <ul class="section-selector-nav-list two"
                    style="height:30%">

                    </ul>
                </ul>
            </div>
        </div>

        </div>
    </div>

    <div class="vertical-separator" style="height:85%"></div>

    <div class="section-selector-content">
        <div class="section-selector-list" id="section_selector_list">
            </div>
        </div>
    </div>
</div>

<div class="site-builder">
    <link href="{{url_for('static', filename='css/default_content_style.css')}}" rel="stylesheet" type="text/css" />
    <div class="site-builder-preview" id="contains_site">

        </div>
    </div>
</div>

<script src="{{url_for('static', filename='js/site-edit.js')}}"></script>

{% endblock %}
```