Linus' Fabulously Tasty cuisine!

Internet Applications, ID1354

Linus Berg and linus@fenix.me.uk

Date

1 Introduction

The task was to create a recipe website that follow common web application design guidelines, and learn HTML & CSS while also learning the design patterns. Page lay-out; font size, family, and style; foreground and background colour; mouse hovering and link behaviour, all of these things had to be explicitly selected, and none may have the default value.

Certain pages were explicitly described for extra clarity.

1.1 Index

This is the front page of the web site, it shall be informative and welcoming. It shall promote the calendar page and have a link to that page.

1.2 Recipes

There shall be one recipe page for each dish. In this first version of the site, there are only two dishes, meatballs and pancakes. A recipe page shall contain the name of the dish, an image of the prepared meal, a list of ingredients, in- structions and user comments. The user shall not be able to write comments, instead you shall hard code sample comments.

1.3 Calendar

The calendar shall be a visual representation of one month, with clickable images of the month's dishes. These images shall be links to corresponding recipes. Your calendar shall have dishes two days in the month, the meatballs day and the pancake day.

2 Literature Study

I was already quite familiar with all three technologies used (Javascript, HTML, CSS), therefor a lot of literature studies could be skipped. However W3C, Jquery API reference, Stackoverflow, and some sites were used for looking up specific commands/tags.

3 Method

Explain how you worked when solving the tasks and how you evaluated that your solution met the requirements. Mention development tools and IDEs you used. *Do not explain your solution and do not refer to code*, that belongs to the *Result* section.
The editor used for writing code was Vim, the rest of the software stack utilised Python, Flask, and running all this was the WSGI server Gunicorn.

Note: If the server was to be facing 'outward' and accept public traffic, Nginx would have been used as a reverse proxy to Gunicorn, however this was not part of the task.

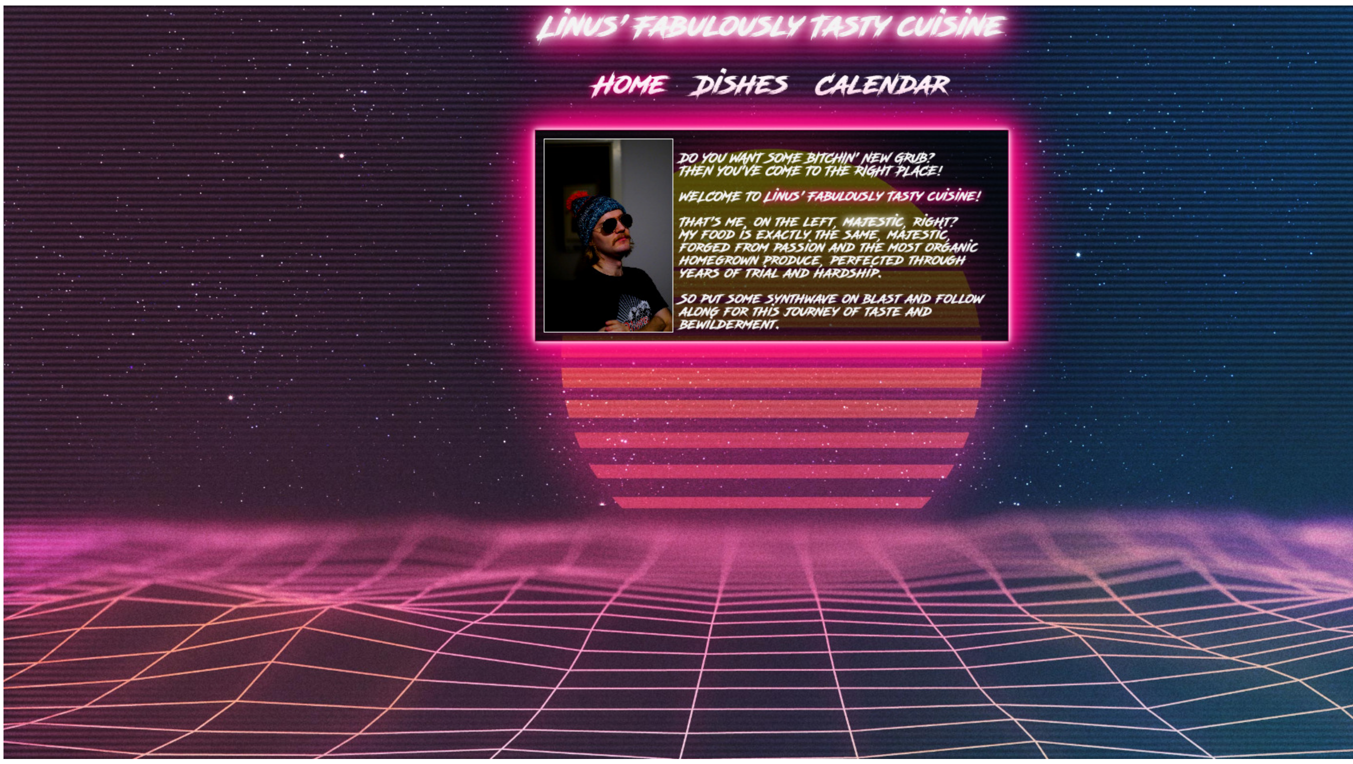
The templating engine Jinja was used to make sure each page follows the same layout / structure, this way was the most simple and frankly involved the least work.

This web stack was chosen because I am familiar with the whole process and my personal opinions is that it is far superior to using a PHP solution as is used further in the course.

The project started out with a simple website that adhered to all the requirements, however it was extremely simple and bland, to avoid making just another website among thousands, I decided to enjoy myself and do something unorthodox. I let the creative juices flow, and made a comedic twist to the website while still displaying the required knowledge for the lab (or at least I hope so).

The development occured iteratively, gathering opinions on fonts, colours, and other design choices, and fixed the issues accordingly. A lot of inspiration was drawn from the 80's Outrun / Synthwave scene. Scaling measurements (rem / %) were also used to ensure website compatability. Javascript was also extensively used to automatically generate HTML code, for easier portability.

4 Result



DO YOU WANT SOME BITCHIN' NEW GRUB?
THEN YOU'VE COME TO THE RIGHT PLACE!

WELCOME TO LINUS' FABULOUSLY TASTY CUISINE!

THAT'S ME, ON THE LEFT, MAJESTIC, RIGHT?
MY FOOD IS EXACTLY THE SAME, MAJESTIC,
FORGED FROM PASSION AND THE MOST ORGANIC
HOMEGROWN PRODUCE, PERFECTED THROUGH
YEARS OF TRIAL AND HARDSHIP.

SO PUT SOME SYNTHWAVE ON BLAST AND FOLLOW
ALONG FOR THIS JOURNEY OF TASTE AND
BEWILDERMENT.

Figure 1: The inviting homepage!

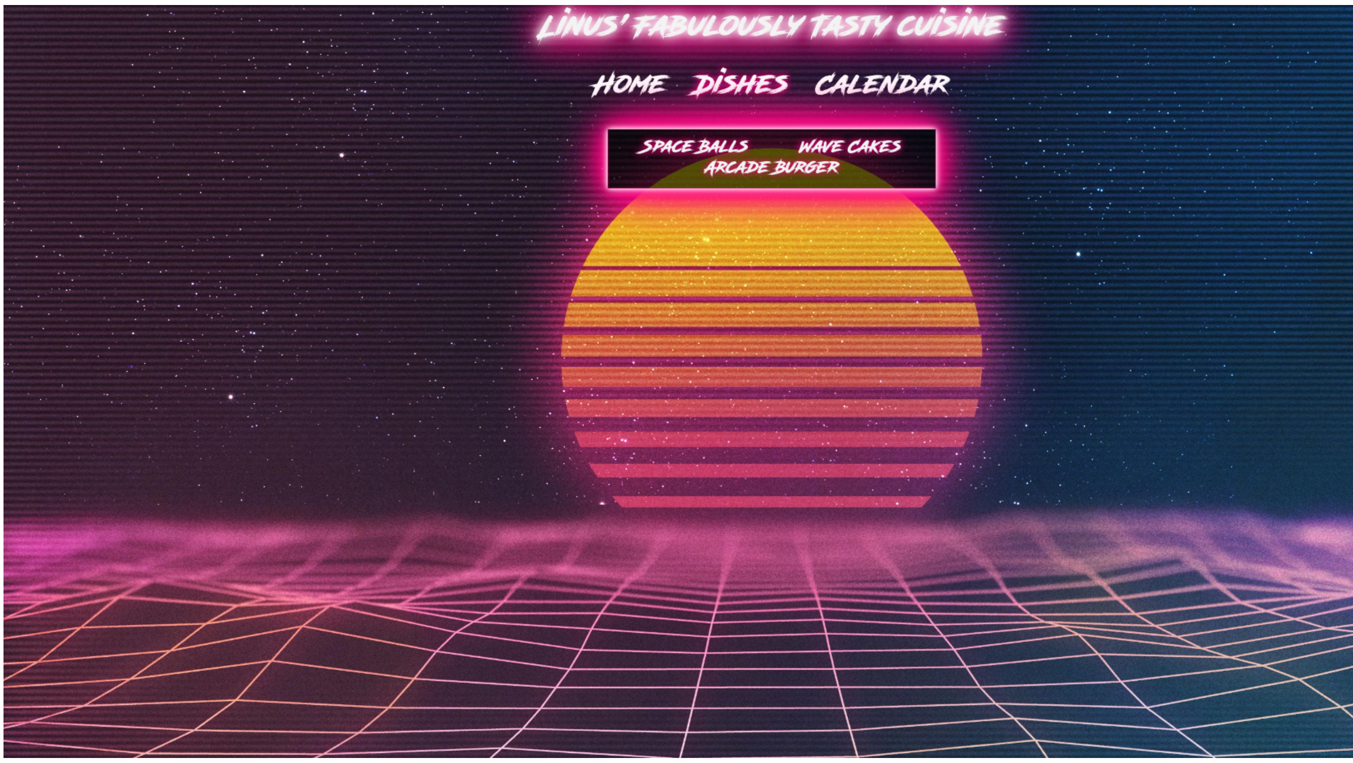


Figure 2: The pages displaying dishes available for the user to pick!

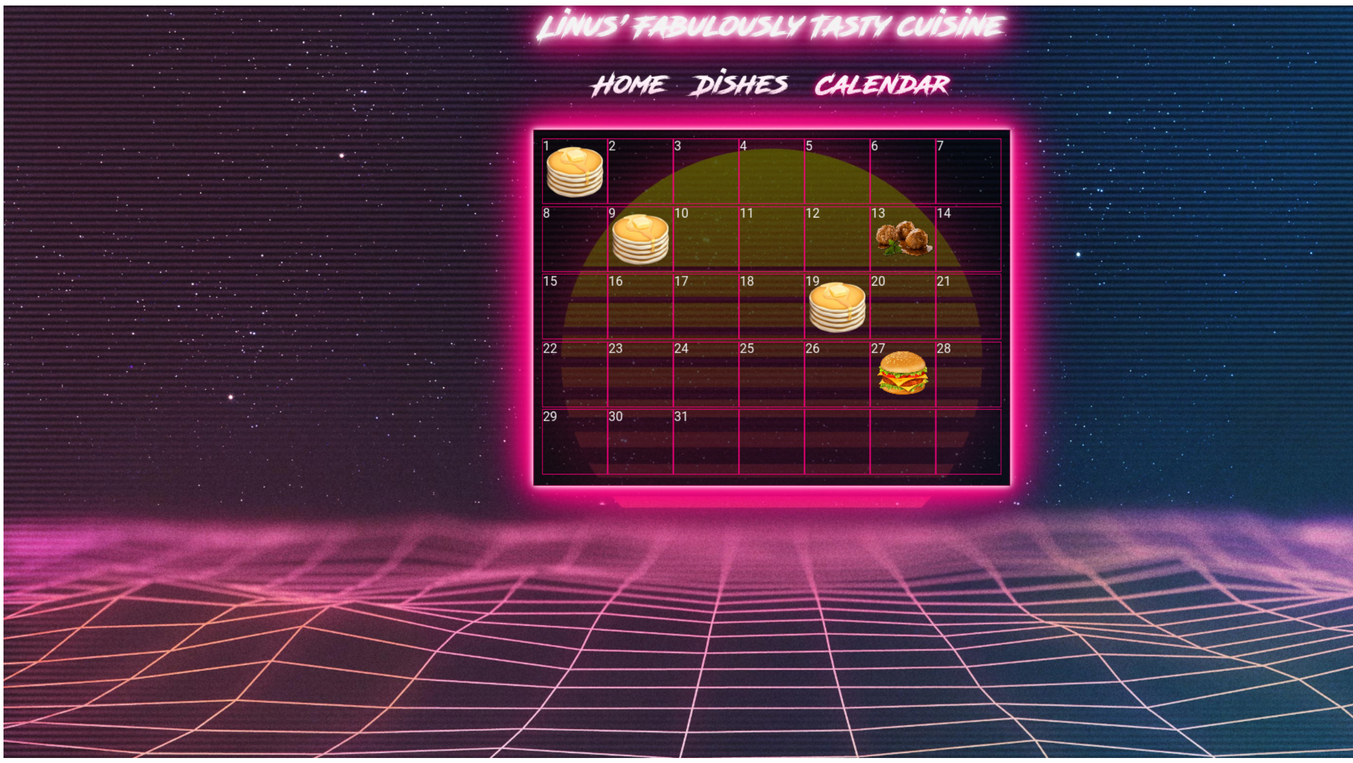
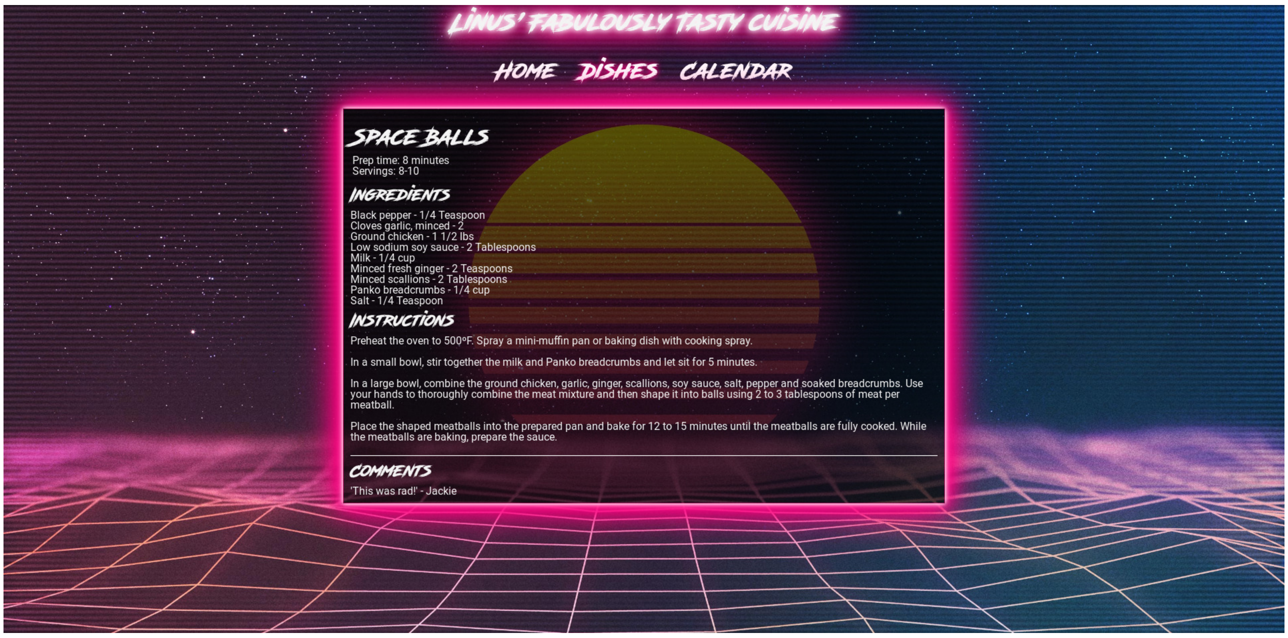


Figure 3: The dish calendar!



SPACE BALLS

Prep time: 8 minutes
Servings: 8-10

INGREDIENTS

Black pepper - 1/4 Teaspoon
Cloves garlic, minced - 2
Ground chicken - 1 1/2 lbs
Low sodium soy sauce - 2 Tablespoons
Milk - 1/4 cup
Minced fresh ginger - 2 Teaspoons
Minced scallions - 2 Tablespoons
Panko breadcrumbs - 1/4 cup
Salt - 1/4 Teaspoon

INSTRUCTIONS

Preheat the oven to 500ºF. Spray a mini-muffin pan or baking dish with cooking spray.

In a small bowl, stir together the milk and Panko breadcrumbs and let sit for 5 minutes.

In a large bowl, combine the ground chicken, garlic, ginger, scallions, soy sauce, salt, pepper and soaked breadcrumbs. Use your hands to thoroughly combine the meat mixture and then shape it into balls using 2 to 3 tablespoons of meat per meatball.

Place the shaped meatballs into the prepared pan and bake for 12 to 15 minutes until the meatballs are fully cooked. While the meatballs are baking, prepare the sauce.

COMMENTS

'This was rad!' - Jackie

Figure 4: The recipe specification with comments!

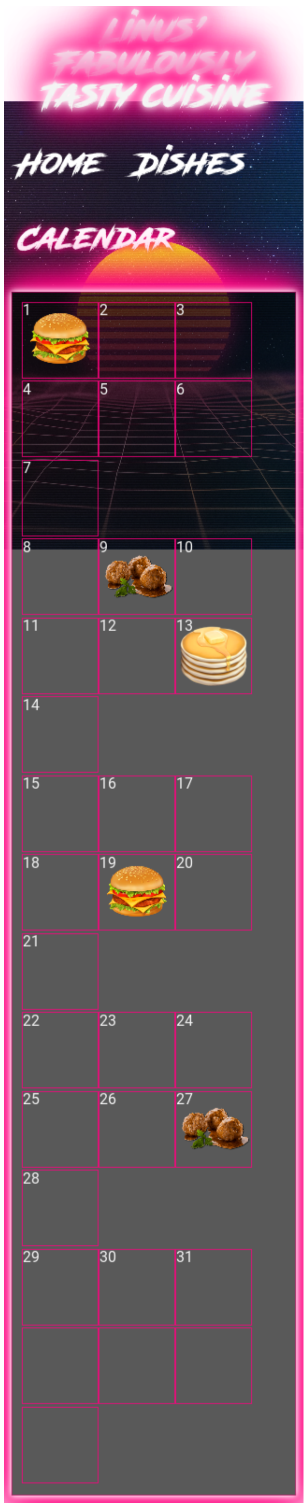
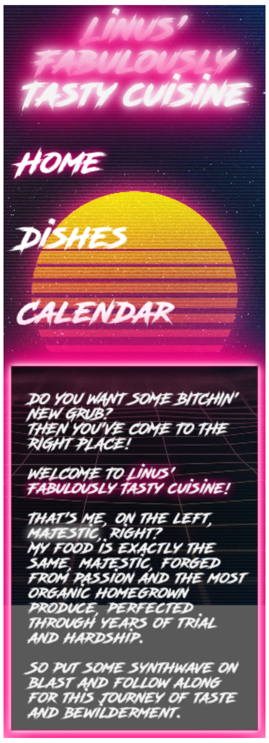


Figure 5: Example of mobile scaling!



DO YOU WANT SOME BITCHIN' NEW GRUB?
THEN YOU'VE COME TO THE RIGHT PLACE!

WELCOME TO LINUS' FABULOUSLY TASTY CUISINE!

THAT'S ME, ON THE LEFT, MAJESTIC, RIGHT?
MY FOOD IS EXACTLY THE SAME, MAJESTIC, FORGED FROM PASSION AND THE MOST ORGANIC HOMEGROWN PRODUCE, PERFECTED THROUGH YEARS OF TRIAL AND HARDSHIP.

SO PUT SOME SYNTHWAVE ON BLAST AND FOLLOW ALONG FOR THIS JOURNEY OF TASTE AND BEWILDERMENT.

Figure 6: Example of mobile scaling!

Github

The website should meet most of the requirements, however with own design choices implemented here and there, such as no image is present in the specification of the recipe, however it is present in the calendar, this was a concious choice. The following design patterns are however met.

4.1 Visibility of system status

This is not really that relevant to the website, however it conveys where the user is currently browsing on the site by highligthing the navigation bar, see Figure **??**.

4.2 Match between system and the real world

No confounding language is used except some rad 80s slang (see Figure **??**), the only thing that may be confusing for a user was the renaming of common dishes to more align with the website theme.

4.3 Consistency and standards

The website has consistent naming of variables / classes / ids and pages, it also adheres to the W3C standard (validation by the W3C validator was achieved).

4.4 Recognition rather than recall

The website is so simple the ordinary user should have no issues recognizing where to go, and the project is too small to really drive this point home.

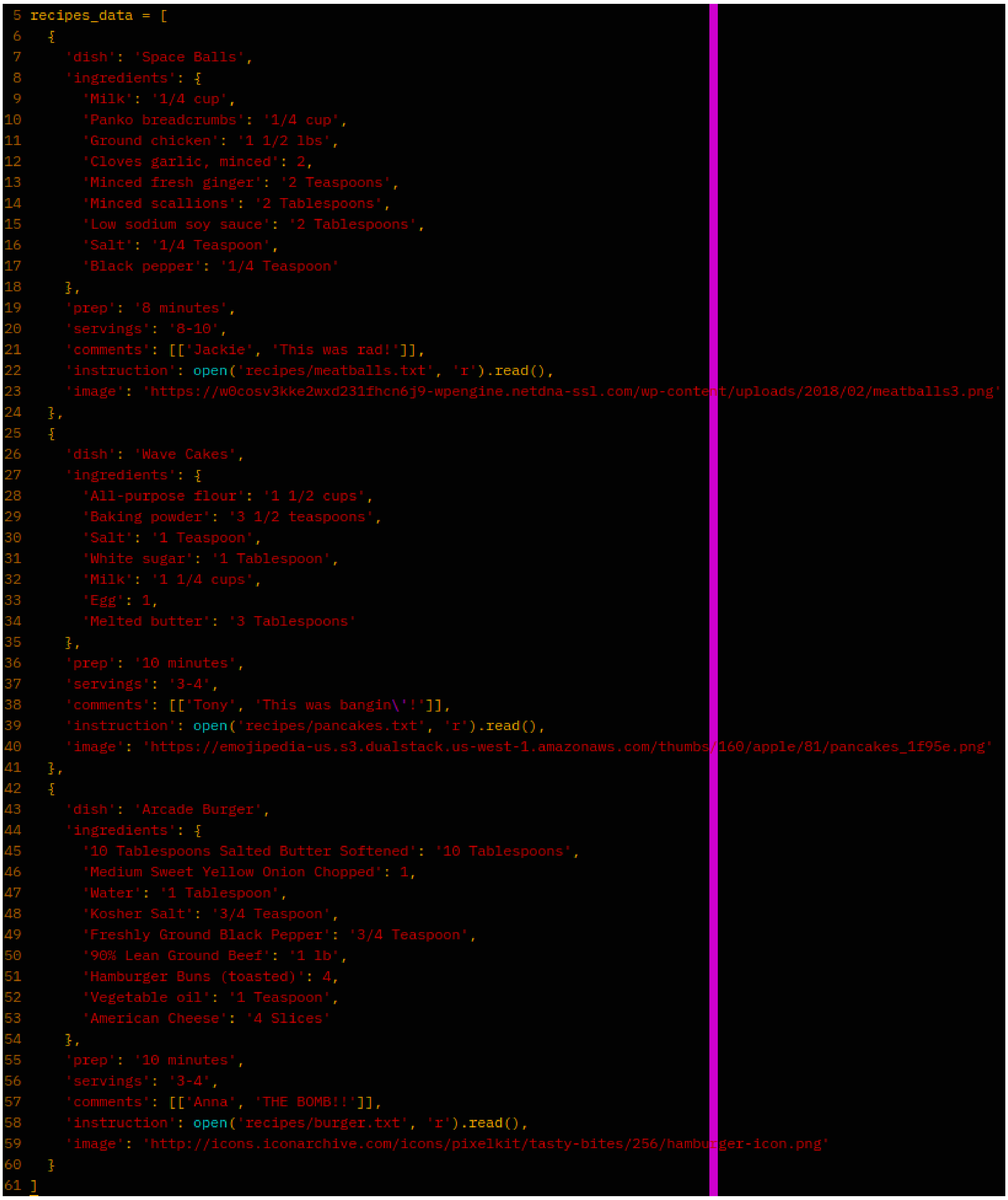
4.5 Aesthetic and minimalist design

This is the requirement the website strays from the norm the most, however even though the design is quite outrageous, it is still quite aesthetically pleasing and minimalistic (one of the reasons images was removed from the recipe specification). The calendar however does not look that good and does not scale well, see Figure **??**.

The website was tested in several browsers and mobile devices, such as Firefox, Chrome, and Edge, IE11 support is average at best, however it still displays, somewhat properly. The site performs well on mobile devices as well, scaling decently, however some pages could use some more work. Overall the result is pleasing and light-hearted.

4.6 Code

The code utilises HTML, CSS, Python, and Javascript, these combined makes less repetion of hardcoded values and templates such as the recipe page.

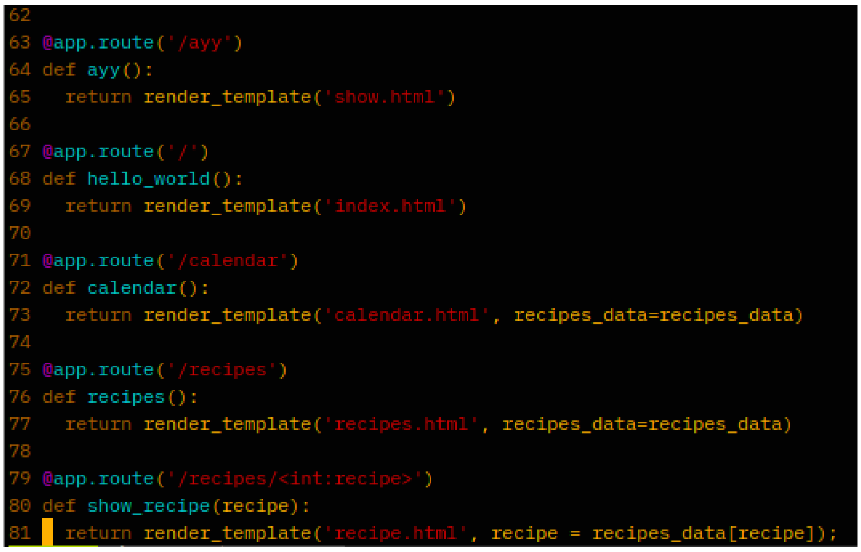


```
recipes_data = [
  {
    'dish': 'Space Balls',
    'ingredients': {
      'Milk': '1/4 cup',
      'Panko breadcrumbs': '1/4 cup',
      'Ground chicken': '1 1/2 lbs',
      'Cloves garlic, minced': 2,
      'Minced fresh ginger': '2 Teaspoons',
      'Minced scallions': '2 Tablespoons',
      'Low sodium soy sauce': '2 Tablespoons',
      'Salt': '1/4 Teaspoon',
      'Black pepper': '1/4 Teaspoon'
    },
    'prep': '8 minutes',
    'servings': '8-10',
    'comments': [['Jackie', 'This was rad!']],
    'instruction': open('recipes/meatballs.txt', 'r').read(),
    'image': 'https://w0cosv3kke2wxd231fhcn6j9-wpengine.netdna-ssl.com/wp-content/uploads/2018/02/meatballs3.png'
  },
  {
    'dish': 'Wave Cakes',
    'ingredients': {
      'All-purpose flour': '1 1/2 cups',
      'Baking powder': '3 1/2 teaspoons',
      'Salt': '1 Teaspoon',
      'White sugar': '1 Tablespoon',
      'Milk': '1 1/4 cups',
      'Egg': 1,
      'Melted butter': '3 Tablespoons'
    },
    'prep': '10 minutes',
    'servings': '3-4',
    'comments': [['Tony', 'This was bangin\'!']],
    'instruction': open('recipes/pancakes.txt', 'r').read(),
    'image': 'https://emojipedia-us.s3.dualstack.us-west-1.amazonaws.com/thumbs/160/apple/81/pancakes_1f95e.png'
  },
  {
    'dish': 'Arcade Burger',
    'ingredients': {
      '10 Tablespoons Salted Butter Softened': '10 Tablespoons',
      'Medium Sweet Yellow Onion Chopped': 1,
      'Water': '1 Tablespoon',
      'Kosher Salt': '3/4 Teaspoon',
      'Freshly Ground Black Pepper': '3/4 Teaspoon',
      '90% Lean Ground Beef': '1 lb',
      'Hamburger Buns (toasted)': 4,
      'Vegetable oil': '1 Teaspoon',
      'American Cheese': '4 Slices'
    },
    'prep': '10 minutes',
    'servings': '3-4',
    'comments': [['Anna', 'THE BOMB!!']],
    'instruction': open('recipes/burger.txt', 'r').read(),
    'image': 'http://icons.iconarchive.com/icons/pixelkit/tasty-bites/256/hamburger-icon.png'
  }
]
```

Figure 7: Example Json structure!

Each recipe was structured as in **??** to allow for easy addition of new recipes. These were simply hardcoded in python to allow for testing.

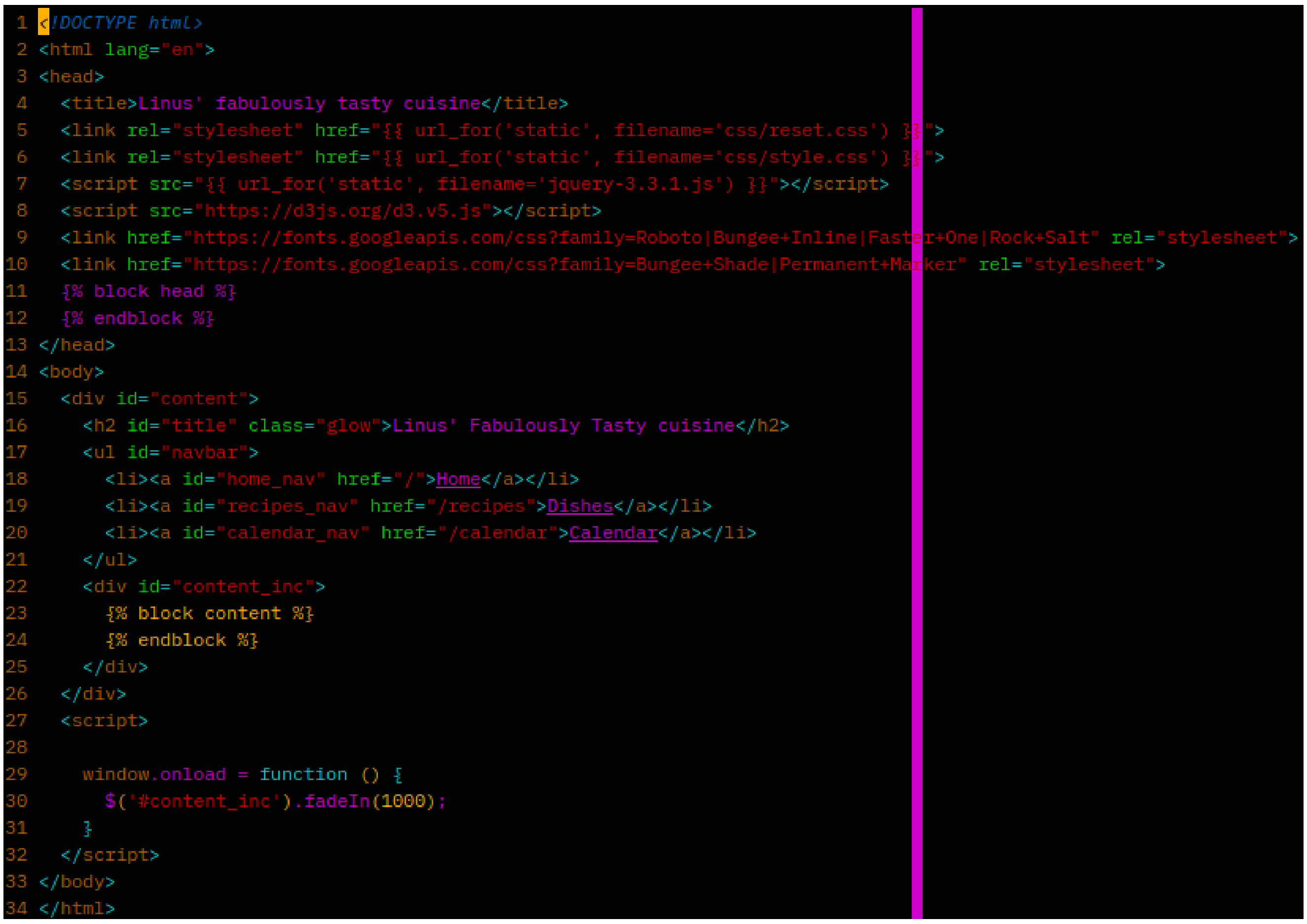
The website uses Python with the Flask framework, this means easy templating and routing for the programmer.



```

@app.route('/ayy')
def ayy():
  return render_template('show.html')

@app.route('/')
def hello_world():
  return render_template('index.html')

@app.route('/calendar')
def calendar():
  return render_template('calendar.html', recipes_data=recipes_data)

@app.route('/recipes')
def recipes():
  return render_template('recipes.html', recipes_data=recipes_data)

@app.route('/recipes/<int:recipe>')
def show_recipe(recipe):
  return render_template('recipe.html', recipe = recipes_data[recipe]);
```

Figure 8: Example routing structure!

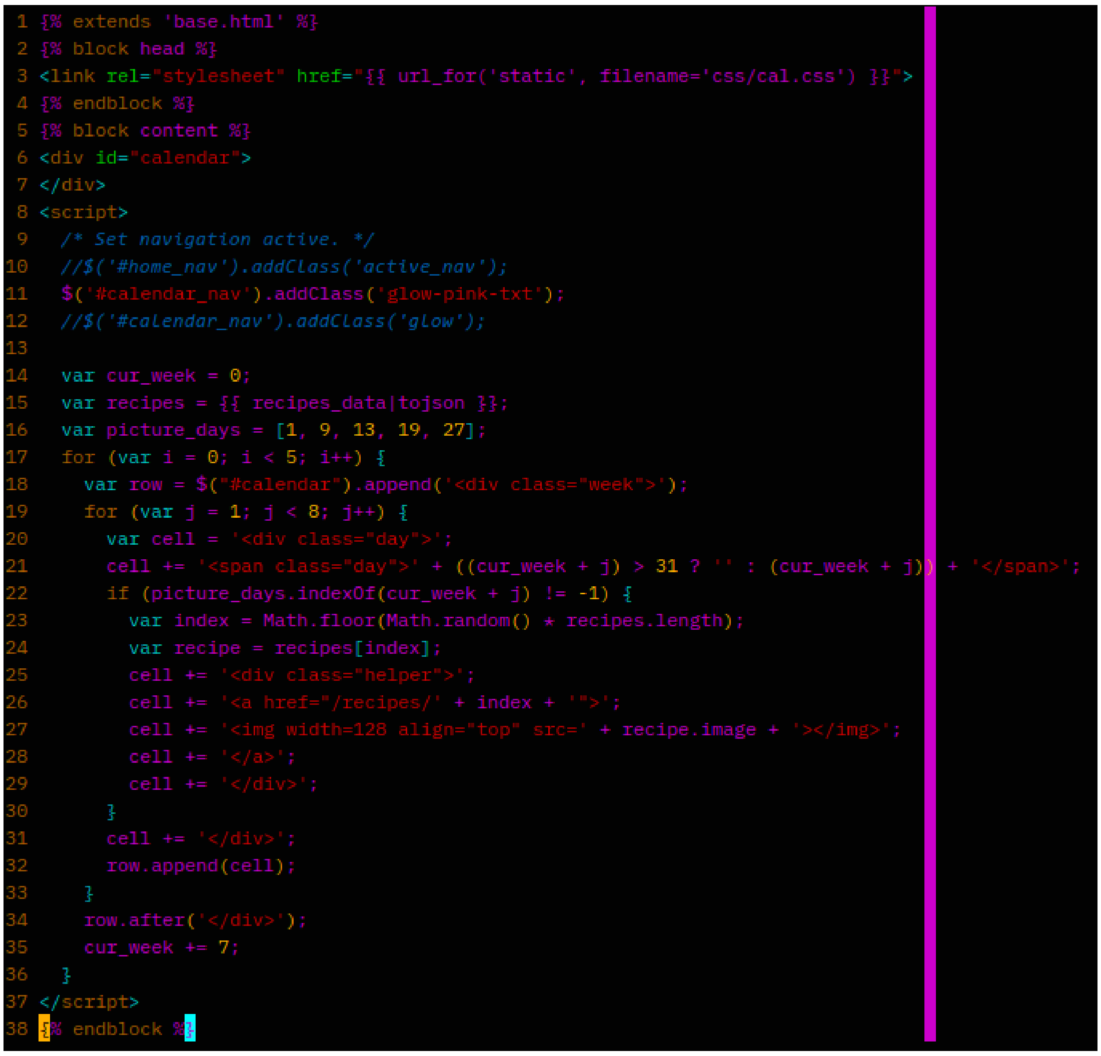
The framework also allows us to pass data between the site and python, as shown in **??** with *recipes_data*.



```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Linus' fabulously tasty cuisine</title>
  <link rel="stylesheet" href="{{ url_for('static', filename='css/reset.css') }}">
  <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
  <script src="{{ url_for('static', filename='jquery-3.3.1.js') }}"></script>
  <script src="https://d3js.org/d3.v5.js"></script>
  <link href="https://fonts.googleapis.com/css?family=Roboto|Bungee+Inline|Faster+One|Rock+Salt" rel="stylesheet">
  <link href="https://fonts.googleapis.com/css?family=Bungee+Shade|Permanent+Marker" rel="stylesheet">
  {% block head %}
  {% endblock %}
</head>
<body>
  <div id="content">
    <h2 id="title" class="glow">Linus' Fabulously Tasty cuisine</h2>
    <ul id="navbar">
      <li><a id="home_nav" href="/">Home</a></li>
      <li><a id="recipes_nav" href="/recipes">Dishes</a></li>
      <li><a id="calendar_nav" href="/calendar">Calendar</a></li>
    </ul>
    <div id="content_inc">
      {% block content %}
      {% endblock %}
    </div>
  </div>
  <script>

    window.onload = function () {
      $('#content_inc').fadeIn(1000);
    }
  </script>
</body>
</html>
```

Figure 9: Example template!

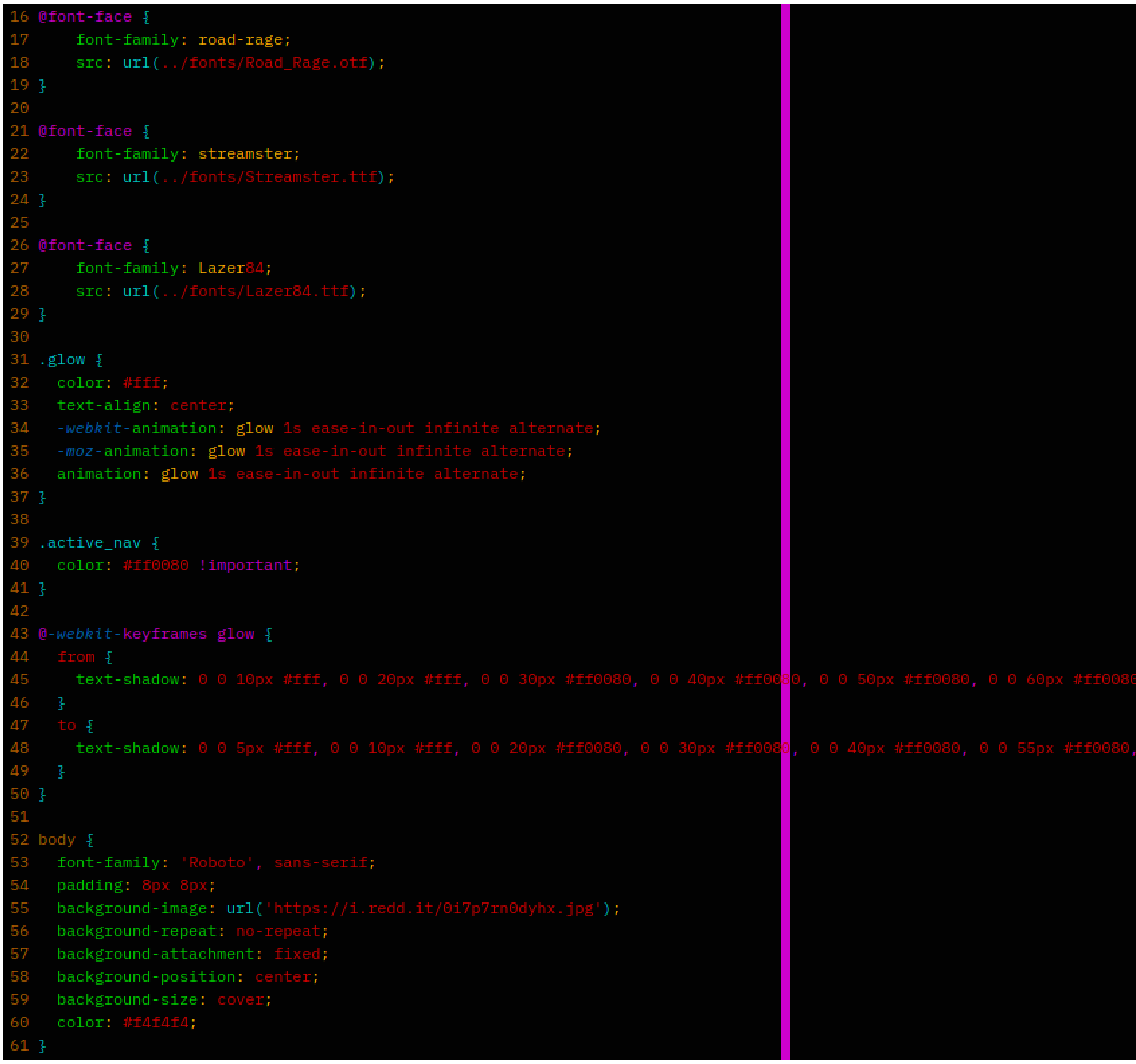
The templating engine Jinja allows us to access the data provided in **??** easily by simple commands, example: '{{ *recipes_data* }}'. The templating engine also allows us to create a 'standard' template for each page, and only change the content block on each, this makes development a lot quicker and easier, especially to give a uniform look to the website as shown in **??**.



```
{% extends 'base.html' %}
{% block head %}
<link rel="stylesheet" href="{{ url_for('static', filename='css/cal.css') }}">
{% endblock %}
{% block content %}
<div id="calendar">
</div>
<script>
  /* Set navigation active. */
  //$('#home_nav').addClass('active_nav');
  $('#calendar_nav').addClass('glow-pink-txt');
  //$('#calendar_nav').addClass('glow');

  var cur_week = 0;
  var recipes = {{ recipes_data|tojson }};
  var picture_days = [1, 9, 13, 19, 27];
  for (var i = 0; i < 5; i++) {
    var row = $("#calendar").append('<div class="week">');
    for (var j = 1; j < 8; j++) {
      var cell = '<div class="day">';
      cell += '<span class="day">' + ((cur_week + j) > 31 ? '' : (cur_week + j)) + '</span>';
      if (picture_days.indexOf(cur_week + j) != -1) {
        var index = Math.floor(Math.random() * recipes.length);
        var recipe = recipes[index];
        cell += '<div class="helper">';
        cell += '<a href="/recipes/' + index + '">';
        cell += '<img width=128 align="top" src=' + recipe.image + '></img>';
        cell += '</a>';
        cell += '</div>';
      }
      cell += '</div>';
      row.append(cell);
    }
    row.after('</div>');
    cur_week += 7;
  }
</script>
{% endblock %}
```

Figure 10: Example javascript!

Javascript was used extensively to generate the website, as shown in ??, the figure shows the generation of the calendar using javascript.



```
@font-face {
    font-family: road-rage;
    src: url(../fonts/Road_Rage.otf);
}

@font-face {
    font-family: streamster;
    src: url(../fonts/Streamster.ttf);
}

@font-face {
    font-family: Lazer84;
    src: url(../fonts/Lazer84.ttf);
}

.glow {
  color: #fff;
  text-align: center;
  -webkit-animation: glow 1s ease-in-out infinite alternate;
  -moz-animation: glow 1s ease-in-out infinite alternate;
  animation: glow 1s ease-in-out infinite alternate;
}

.active_nav {
  color: #ff0080 !important;
}

@-webkit-keyframes glow {
  from {
    text-shadow: 0 0 10px #fff, 0 0 20px #fff, 0 0 30px #ff0080, 0 0 40px #ff0080, 0 0 50px #ff0080, 0 0 60px #ff008
  }
  to {
    text-shadow: 0 0 5px #fff, 0 0 10px #fff, 0 0 20px #ff0080, 0 0 30px #ff0080, 0 0 40px #ff0080, 0 0 55px #ff0080,
  }
}

body {
  font-family: 'Roboto', sans-serif;
  padding: 8px 8px;
  background-image: url('https://i.redd.it/0i7p7rn0dyhx.jpg');
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-position: center;
  background-size: cover;
  color: #f4f4f4;
}
```

Figure 11: Example css!

CSS was also utilised extensively as shown above, several custom fonts were defined via the *@font-face* command. Several CSS3 animations were also used to give the website a more alive look.

5 Discussion

I believe I met all the requirements with Aesthetic and minimalist design being the one failure. The only problem I faced was laziness, for example, the calendar and other pages do not scale brilliantly, especially the calendar, overall I believe I only spent around 4 hours building the site.

The design could also have been better, a better / more fitting font could have been found for the text on the front page, however this would require extensively searching for a fitting font, something I deemed unecessesary for the task.

The report template also states the code should be explained, however I found it diffcult on these tasks to actually explain, as I am not sure which parts to explain.

I could also have made a bland site for pure functionality, but where is the fun in that?