Web 班活動報告書

立命館コンピュータクラブ
2021 年度プロジェクト活動

2021 年 8 月 19 日

1 回生: hane，澤谷 祐樹，竹中 萌，本多 峰之
2 回生: husky，kawai，sno2wman，川崎 秀昌，きょうすけ，クマさん，仁
3 回生: FKD，OK，TNB，阿部 健太朗，宇佐 基史，中尾 龍矢，星名 藍乃介，やぎちゃん

目次

1　活動概要

文責: 星名 藍乃介

Web 班は，Web の仕組みや構成技術についての知見と実装力を身につけることを目標とし，班員間で Web サービスや開発手法について共有しながら，個人制作物を開発するプロジェクトである．

活動前半では，Web 開発時に有用なリファレンスや教材の共有，初学者のためのチュートリアルや World Wide Web の仕組みについての共有を行った．なお，活動成果の詳細は外部 Web サービス Scrapbox（https://scrapbox.io/rcc-web2021/）でまとめられているので，本報告書で掲載しきれなかった内容に関してはそちらを参照されたい．上記の共有事項以外にも，班員による Web 開発に関する知見がまとめられている．

活動後半では，班員による個人制作物の開発を行った．成果物については後述する．

2　プロジェクトスケジュール

Web 班のプロジェクトスケジュールを，表 1 に示す．

表 1　プロジェクトスケジュール

時期	活動内容
第一回	Scrapbox 企画の説明，教材紹介，環境構築
第二回	目標成果物の決定，Web の仕組み超入門，HTML/CSS/JS チュートリアル
第三回	成果物開発開始，Web フロントエンド技術の歴史，バックエンドチュートリアル
第四回	成果物開発，Web フロントエンドチュートリアル
第五回	成果物開発
第六回	同上
第七回	同上

3　制作物

以下に班員の個人制作物について記載する．

3.1　クロスサイトスクリプティング (XSS) について

文責: hane

3.1.1 概要

クロスサイトスクリプティング（以下，XSS）とは Web アプリケーションの脆弱性を利用した攻撃方法の一つである．今回の活動では XSS について調べた後に脆弱性のある Web アプリケーションを作成し XSS 攻撃を行った．

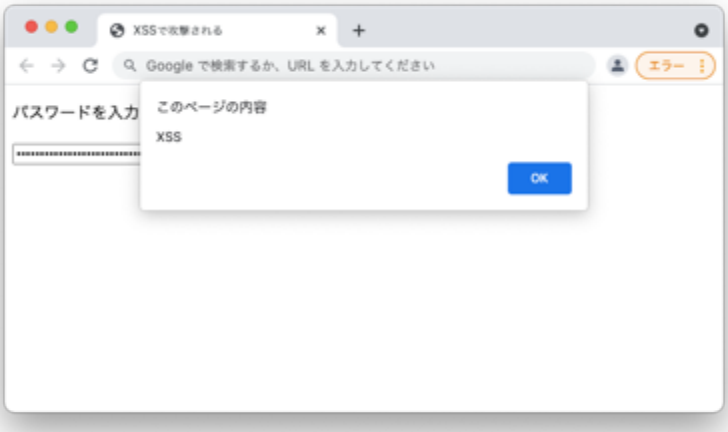
3.1.2 活動内容

作成した Web アプリケーションはパスワードを入力し送信すると，入力したパスワードが HTML 内で表示されるものである．作成した Web アプリケーションでは `innerHTML` を使って要素を生成しているが HTML エスケープをしていないので，送信欄に`<img src onerror="alert(‘XSS‘)">` を入力し送信するとブラウザ上に表示される．この脆弱性を利用すると Web ページ内の機密情報を手に入れることができる．

3.1.3 苦労した点

XSS を理解するにあたって HTML や JavaScript といったプログラミング言語を習得する必要があった．

3.1.4 スクリーンショット



3.2 ポートフォリオサイト

文責: 澤谷 祐樹

3.2.1 概要

自身のポートフォリオである．筆者自身についてのページ，普段利用している discord のサーバーへのリンク，Twitter にリンクした画像や絵を並べたページを作った．

3.2.2 URL

https://kaze8888.github.io/kazeportfolio/

3.2.3 使用技術

HTML, CSS

3.2.4 工夫した点

なるべくシンプルにするために使う色を少なくした．具体的には，About と Gallery の部分で使う色を 1 種類にし，灰色を使用することによって色が目立つようにしたのが工夫した点である．

3.2.5 苦労した点

フォントを選ぶのが苦労した点である．特に，大きい文字に使うフォントは見つけるのに苦労した．また，レスポンシブに対応させる際，margin を多用していたため，対応させるのに時間がかかり苦労した．

3.2.6 スクリーンショット

3.3 BTS ファンページ

文責: 竹中 萌

3.3.1 概要

韓国の 7 人組男性ヒップホップグループである BTS のメンバーについて，簡単に紹介したものである．

3.3.2 使用技術

HTML, CSS

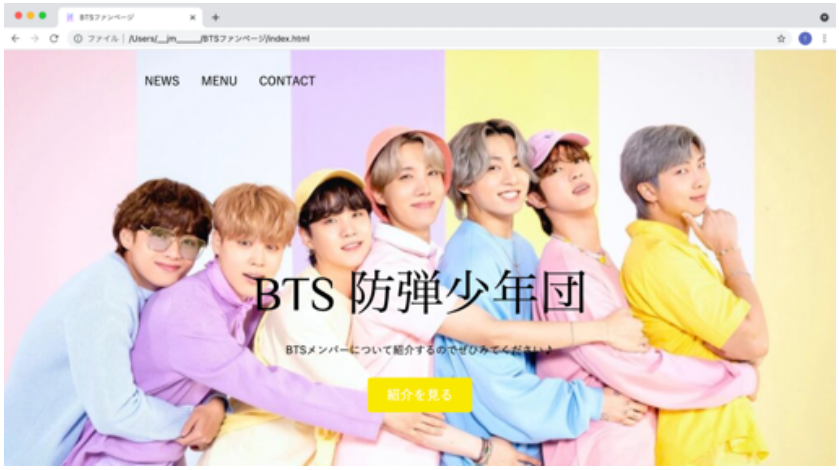
3.3.3 工夫した点

筆者のデザインにおける知識量を考慮し，華やかな色の画像をバックグラウンドに使うことで，全体的なバランスを補った点．

3.3.4 苦労した点

HTML，CSS の全体的な仕組みや，各タグがどのように反映されるのかについての理解．

3.3.5 スクリーンショット



背景画像の出典[*1]

favicon の出典[*2]

3.4 チャットアプリ（web）

文責: 本多 峰之

3.4.1 概要

Udemy で React 初学者に対して，チャットアプリの開発の説明があった[*3] のでそれを視聴しながら作成を行った．今回チャットアプリの開発を行ったのは，高校の時に考えていたアプリを作りたいという思いからである．最終的にはコワーキングスペースの中で新たな価値のあるビジネスを創造するため，様々な職種の人がマッチングできるサービスを作りたいと考えている．そのためにはチャット機能が必要であったため，どのように動いているのか，仕組みを調べながら進めることにした．

3.4.2 使用技術

- Next.js
- React
- firebase

[*1] BTS 防弾少年団【情報サイト】.「2021 BTS FESTA のオープニングセレモニー「8 周年家族写真」がついに公開！！」. https://bts613-bighit.com/2021-bts-festa-opening-ceremony/

[*2] Icons8.「グラデーションライン スタイルでの BTS のアイコン」. https://icons8.jp/icon/78538/bts

[*3] Udemy：Firebase 未経験者のための React で作るチャットアプリ開発入門！最速最短でゴール到達！ https://www.udemy.com/course/react-firebase/

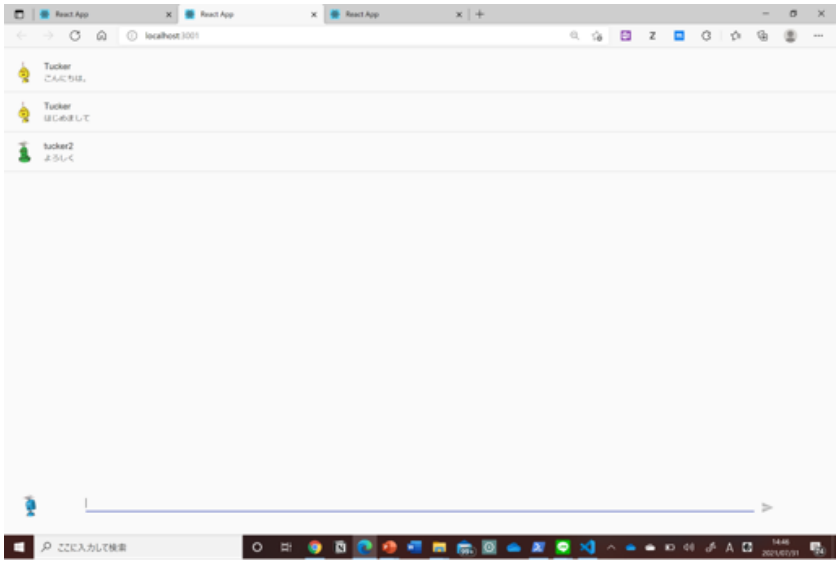
3.4.3 苦労した点

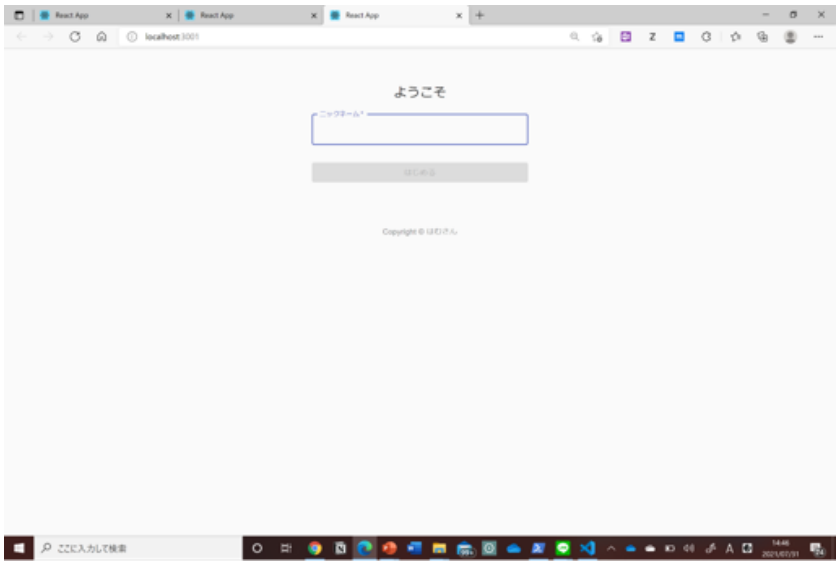
説明がすべて macOS で行われていたため，windowsOS に変換しながら環境構築やコーディングを進めるのが大変であった．また多くのコンポーネントを用いて作成するため，どれがどのコンポーネントを読みとって動いているというのを確認しながらの作業が難しかった．

3.4.4 今後の展望

現在はまだ Udemy で見たものを同じように作成したため，独創性に乏しいと考える．そのため，UX を考え，よりチャットアプリに近づけるように進化させていきたいと思う．

3.4.5 スクリーンショット





3.5 Numer0n

文責: husky

3.5.1 概要

ヌメロンという数字当てゲームを JavaScript で制作した

3.5.2 URL

https://husky21st.github.io/numeron/

3.5.3 使用技術

- Next.js
- React

3.5.4 工夫した点

JSX を用いて Web 開発ができた.

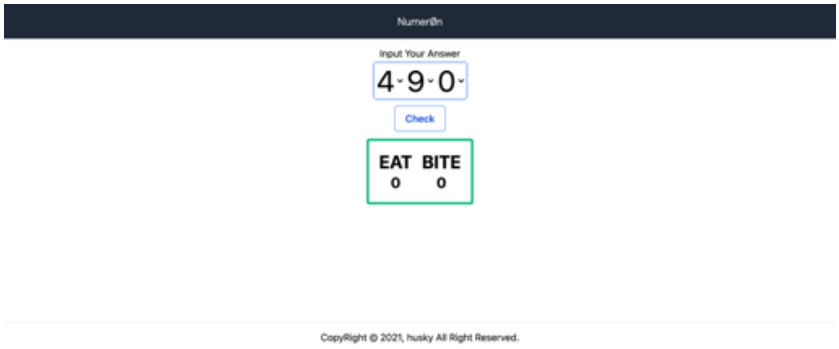
3.5.5 苦労した点

DOM 操作全般．特に，hook の機能と非同期処理の理解に手間取った.

3.5.6 今後の展望

pixi.js 等を使い動きのあるサイトを作っていきたい.

3.5.7 スクリーンショット



3.6 EC サイト

文責: kawai

3.6.1 概要

個人で作った web サービス，モバイルアプリ，ブログアカウントなどが売買できるサービス

3.6.2 使用技術

- Rails
- React
- Docker
- Stripe
- AWS

3.6.3 工夫した点

技術選定（JavaScript ライブラリの選択など）．車輪の再発明をしないために，いいライブラリがあればそれを使うようにした．

3.6.4 苦労した点

Stripe のドキュメントを読み，必要な API を使うこと．

3.6.5 今後の展望

Stripe の導入を完了させて，あとは UI を中心に，細かいバグなどにも対応していく．人のためになるサービスにする．

3.7 henken.club

文責: sno2wman

3.7.1 概要

偏見本棚[*4]という概念があり，これをソーシャル Web サービスとしてプラットフォームを提供できれば面白そうと思い開発している．十分なデータが集まれば，ある程度の傾向を掴んで精度の高いレコメンドなども行えると期待している．現在プロトタイプの開発中.

3.7.2 URL

- GitHub Org
  - バックエンド (プロトタイプ)
  - フロントエンド (プロトタイプ)

3.7.3 使用技術

現在採用しているが，今後変わる可能性もある.

- Web
  - フレームワーク
    - NextJS
  - デザイン
    - TailwindCSS
    - styled-components
  - テスト
    - Storybook
    - Jest
  - ツール
    - graphql-codegen
- App
  - フレームワーク
    - NestJS
    - Prisma 2
  - API 仕様
    - GraphQL
  - テスト

[*4] 参考：https://hito-horobe.net/articles/henken-hondana

        * Jest
- DB
  - MySQL
  - Neo4j
- CI/CD
  - GitHub Actions
  - Renovate

バックエンド側は Docker イメージとしてビルド，ghcr.io にデプロイが行われている．フロントエンド側は未定．

3.7.4 工夫した点

バックエンド側の DB は MySQL（実際には Prisma 2 で透過的に扱うので何でもいいが）と Neo4j のハイブリッドにした．Neo4j では RDB では扱いづらい，**関係**という概念をスマートに扱うことができる．[*5]ハイブリッドにしたのはユーザー情報などは従来のスキーマのある DB で管理したほうがいい（Neo4j はスキーマレス DB）と思ったからである．なお Neo4j には現状 JavaScript 用のクエリビルダなどは無いので自力でクエリをかなり書く必要がある．

API は GraphQL として提供することにした．ソーシャルサービスではフロント側から API 側に要求する情報が多く，従来の RESTful な API だとデータ量が肥大化する傾向があるからである．またフロント側では，graphql-codegen で API の型定義を生成することが出来て開発効率が良いという利点もある．

フロントエンド側は 2021 年現在モダンな技術スタックで組んだ．モダンな技術が常に良いとは限らないが．

3.7.5 苦労した点

認可周りが滅茶苦茶で，非常に脆弱性のある作りになっていると思う．フロントエンド側は Vercel に任せるなり方法はあると思うが，バックエンド側のデプロイの仕方がわからない．これらに詳しい人がいたらご一報ください．

3.8 lufe Site

文責: 川崎 秀昌

3.8.1 概要

自身の Web サイトである． 4つのページで構成されており，経歴，興味分野，SNS や github 等のアカウントを知ることができる．自分自身を世に発信する，見える制作物を作成したいという動機から制作に至った．

[*5] グラフデータベース ——Neo4j によるグラフデータモデルとグラフデータベース入門など参考．

3.8.2 URL

https://lufe.jp/

3.8.3 使用技術

- Next.js
- React
- tailwindcss

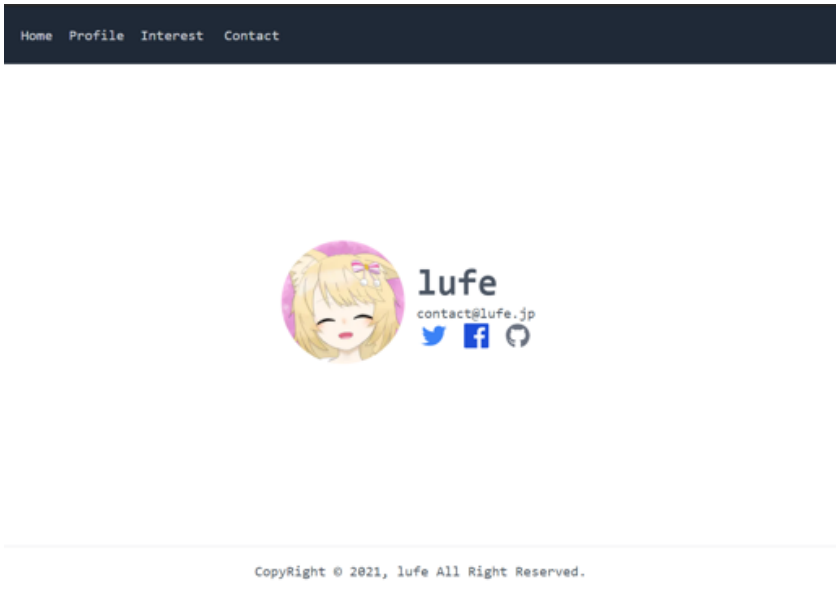
3.8.4 工夫した点

モダンな Web 技術を用いて作成するといったことを念頭に置いたため，現在使われている技術を用いて実装した．

3.8.5 苦労した点

Web ページのレスポンシブデザインに対応させる点が苦労した．

3.8.6 スクリーンショット



3.9 購入申請 API

文責: クマさん

3.9.1 概要

会計局の既存のシステムの保守が難しいため，その更新を目的として今回の API を実装した．実装に当たって保守管理のしやすさを優先した．機能としては備品購入申請の受付，書籍注文，月次決算情報の自動作成といったものを備えている．なお，一部機能は未実装である．

本来であれば，フロントエンドも実装し，会計局のシステムを置き換える予定であったがフロントエンドの実装が間に合わなかったため，今回は API のみとなる．

3.9.2 ソースコード

https://github.com/CityBear3/apply_system

3.9.3 使用技術

- Spring Boot
- Spring Batch
- Kotlin
- MySQL

3.9.4 工夫した点

Java ではなく，Kotlin を用いて実装することにより null 安全や関数型やコルーチンなどのモダンな機能を使用しつつ，シンプルなソースコードを目指した．また，オニオンアーキテクチャをベースとしてドメイン駆動設計を行い，各機能間の結合を疎に保ち保守性を高めた．

初期ユーザーを Spring Batch を用いた非同期バッチ処理によってサーバー起動時に自動作成されるようにした．この処理は一度実行されるとデータベースにメタデータを記録するにことによりサーバーを再起動しても再び実行されることはなく，これによって重複処理を防ぐようにした．

3.9.5 苦労した点

Spring Initializr を用いてプロジェクトの雛形を生成したが依存関係に抜けがあり，サーバーが起動しないというトラブルに手を焼いた．また，Spring Batch についての情報収集に苦労した．

3.9.6 スクリーンショット

3.9.7 API の仕様

機能	URL	method	権限
ログイン	/login	POST	
ログアウト	/logout	POST	
備品購入申請	/order	POST	
パスワード変更	/user/password	POST	
ユーザー作成	/admin/user/register	POST	管理者のみ
ユーザー削除	/admin/user/delete/{id}	DELETE	管理者のみ
ユーザー一覧	/admin/user/list	GET	管理者のみ

その他 API は実装中である.

3.10 Tetris made with React

文責: FKD

3.10.1 URL

https://fukada6280.github.io/react-hooks-tetris/

3.10.2 概要

React を用いて Tetris を作成した.

3.10.3 使用技術

- TypeScript
- React
- Chakra UI

3.10.4 工夫した点

- canvas を用いずに，Hooks で state を更新することによって Tetris を構成した．テトリスのフィールドは Grid で表現している.
- create-react-app の初期配色をメインテーマ色にした点
- サウンドに音割れホッター（http://www.rcc.ritsumei.ac.jp/2020/1210_11055/）を利用した点. BGM は RCC 会員の momonga が制作した，音割れホッターを用いた DTM である.

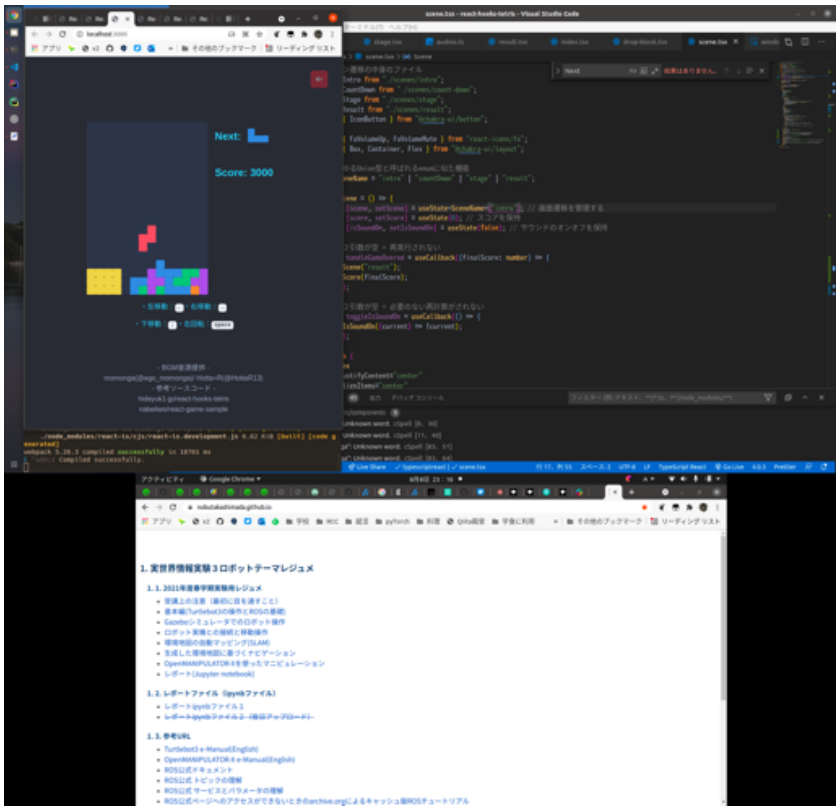
3.10.5　苦労した点

- HTML/CSS の知識が皆無だったために苦労した．HTML を理解していないのに JSX の構文を理解するのは厳しく，CSS を理解していないのに Chakra UI を理解するのは厳しい．
- 関数コンポーネント特有の Hooks を使った状態管理に戸惑った．

3.10.6　今後の展望

- キー入力のレスポンスを改善したい．ブラウザは押しっぱなしによる連続キー入力の判定をあえて遅らせる処理をしている．window を拡張してグローバル変数を保持することでレスポンスが改善すると見られる．
- テトリミノの出現率が極端になる場合があるため，均一化したい．
- 強化学習 AI を搭載してブラウザ上で動かしたい．

3.10.7　スクリーンショット



3.11　ポートフォリオサイト

文責: OK

3.11.1 概要

以前作っていたポートフォリオサイトは HTML と CSS を使用して作成していた．今回はそれをより編集しやすくするために，Next.js と Tailwind CSS で書き換えた．デザインも少し変更した．今後も改善していく．

3.11.2 URL

https://okgwknt.github.io

3.11.3 使用技術

- Next.js
- TypeScript
- Tailwind CSS
- GitHub Pages

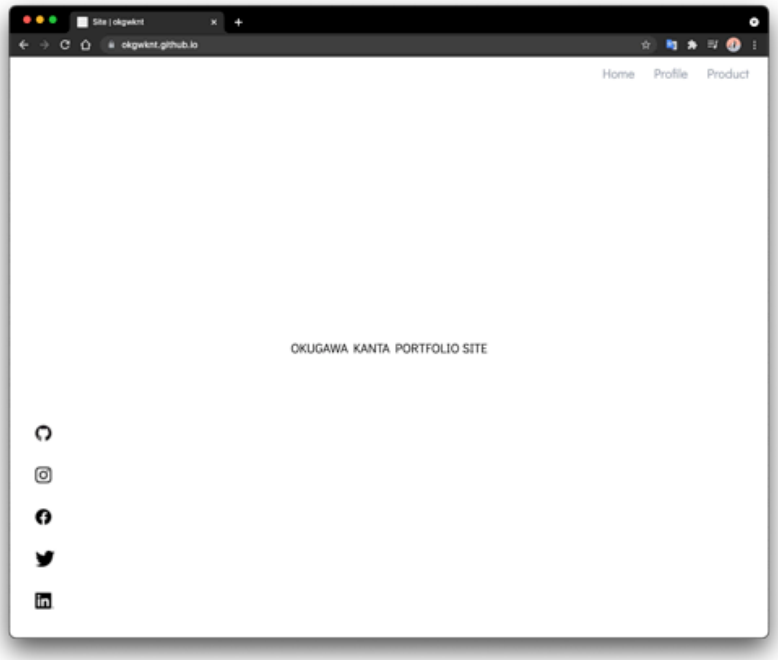
3.11.4 工夫した点

Simple is best の考え方でシンプルなデザインにした．画面の大部分は余白にし，中央の文字列に注目させた．現在はシンプルな文字列となっているが変更する可能性がある．コンポーネントを分けていく作業が面白かった．

3.11.5 苦労した点

JavaScript もあまり触ったことがなかったため，TypeScript を書くのが大変であった．Tailwind CSS は省略名を覚えるのが面倒であった．

3.11.6 スクリーンショット



3.12 RCC 部室予約サービス

文責: 阿部 健太朗

3.12.1 URL

- URL：https://crms.abelab.dev/
- ソースコード：https://github.com/averak/CRMS
- 仕様：https://api.abelab.dev/crms/swagger-ui/

3.12.2 概要

現在 RCC の部室は入室不可だが，大学から対面活動が許可された場合には人数制限をかけた上で利用が可能となる．そこで，いつ，誰が部室を利用するのか予定を管理するサービスを開発した．会員は Google カレンダーに予定を追加する要領で，予約の追加・編集・削除を行い，制限人数に応じた抽選結果が Slack に通知される．

表 2　URI 仕様

名前	URI	権限	
		管理者	一般
トップページ	/	○	○
ログイン	/login	○	○
マイページ	/mypage	○	○
プロフィール更新	/mypage/profile	○	○
パスワード更新	/mypage/password	○	○
ダッシュボード	/dashboard	○	○
予約一覧	/reservations	○	○
活動実績	/activities	○	○
エラーページ	/error	○	○
ユーザ一覧	/admin/users	○	
ユーザ作成	/admin/users/new	○	
ユーザ詳細	/admin/users/${user_id}	○	
ユーザ編集	/admin/users/${user_id}/edit	○	

表 3　内部 API 仕様

名前	URI	Method	権限	
			管理者	一般
ログイン	/api/login	POST	-	-
ユーザ一覧取得	/api/users	GET	○	
ユーザ作成	/api/users	POST	○	
ユーザ更新	/api/users/${user_id}	PUT	○	
ユーザ削除	/api/users/${user_id}	DELETE	○	
ログインユーザ詳細取得	/api/users/me	GET	○	○
ログインユーザ更新	/api/users/me	PUT	○	○
ログインユーザパスワード更新	/api/users/me/password	PUT	○	○
予約一覧取得	/api/reservations	GET	○	○
予約作成	/api/reservations	POST	○	○
予約更新	/api/reservations/${reservation_id}	PUT	○	○
予約削除	/api/reservations/${reservation_id}	DELETE	○	○
予約抽選	/api/batch/reservations/lottery	GET	-	-

3.12.3　使用技術

- フロントエンド

  - Angular
  - Angular Material
- バックエンド
  - Spring boot
  - MySQL
- CI/CD
  - GitHub Actions
  - Jenkins

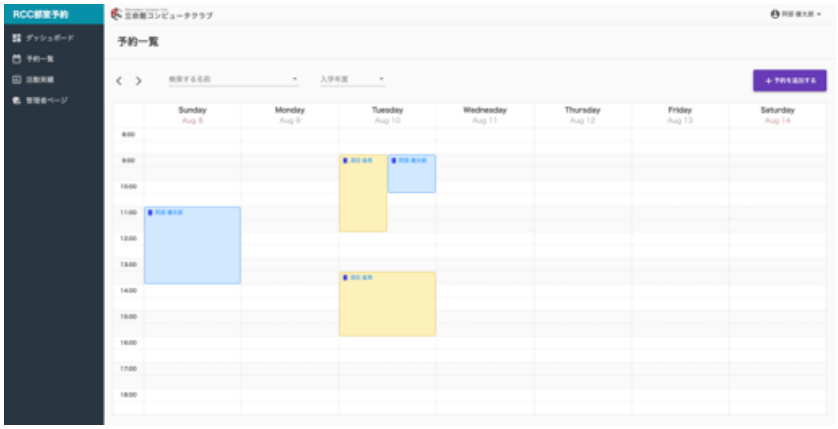
3.12.4　工夫した点

メソッド単位で検証する単体テストと，API エンドポイント単位で想定されるシナリオを検証する統合テストを作成することで，安心かつ保守性の高いプログラムにした．またドメイン駆動設計による責務分割とテスティングファーストの考えを導入したため，ノンストレスで開発することができた．

3.12.5　苦労した点

フロントエンドのコンポーネント分割．

3.12.6　スクリーンショット



3.13　基礎文法解釈の過程の残骸の GitHubPages での公開

文責: 宇佐 基史

3.13.1 概要

参考にしたのは MDN Web Docs[*6]というサイトである．特筆に値するものでもないが，HTML,CSS において基本的なコマンドと思われるものを単純に羅列し，Web 上に公開するまでを目標とした．筆者の学習過程の軌跡である．

3.13.2 URL

- URL：https://thetaleisalsowhite.github.io/
- ソースコード：https://github.com/thetaleisalsowhite/thetaleisalsowhite.github.io

3.13.3 今後の展望

今回の活動で得た知識を基礎にオリジナリティを感じられるモノを実現したい．また，スマートフォン表示におけるこれら言語の仕様に関して自覚を伴うレベルでの理解不足があるので解決に努めたい．

3.13.4 スクリーンショット

期限提出後も更新を続けている．

3.14 立命館大学交響楽第一回オペラ公演公開までのカウントダウンの掲載

文責: 宇佐 基史

3.14.1 URL

https://www.ruso60.com/

[*6] https://developer.mozilla.org/ja/

3.14.2　概要

立命館大学交響楽団の方式 HP は管理において Jimdo という Web ホスティングサービスを用いているが，一部対応外の機能においては HTML に直接書き込む形で実装が可能である．カウントダウンタイマーの実装を経ることで基本的な JavaScript の仕様の理解を目指した．

3.14.3　苦労した点

作業時間の捻出

3.14.4　今後の展望

今回の活動で得た知識を基礎にオリジナリティを感じられるモノを実現したい．また，スマートフォン表示におけるこれら言語の仕様に関して自覚を伴うレベルでの理解不足があるので解決に努めたい．

3.14.5　スクリーンショット

期限提出後も更新を続けている．カウントダウンタイマーについて，デザインの面で爪の甘さを感じているが，楽団内からは比較的好評である．現状の表示では確認できないが，カウントダウン終了後メッセージが表示される予定である．チケット買ってね!!

3.15　ダウンジングゲーム

文責: 中尾 龍矢

3.15.1　URL

- URL：https://downsing.netlify.app/
- ソースコード：https://github.com/Ta-2/dowsing

3.15.2 概要

HTML の canvasJjavaScript の画面幅やディスプレイの大きさをを取得する関数を用いて，ブラウザ上でダウンジングゲームが出来る．

1. プログラム起動
2. 正方形の新しいウィンドウが出現
3. そのウィンドウを移動して，レーダー等から埋蔵物の位置を推理する
4. 任意のキーを押して位置を決定，埋蔵物との距離に近ければ近いほど加算スコアが高くなる．
    - 画面中央の赤い円はレーダーであり，埋蔵物との距離によって反応する．
    - そのほかに小さな黄色い円が埋蔵物の周辺に出現し，埋蔵物の位置の推理に有効活用出来る．

※画面の大きさを固定にして遊ぶため，ポップアップの通知は許可すること

3.15.3 使用技術

- HTML
- CSS
- JavaScript

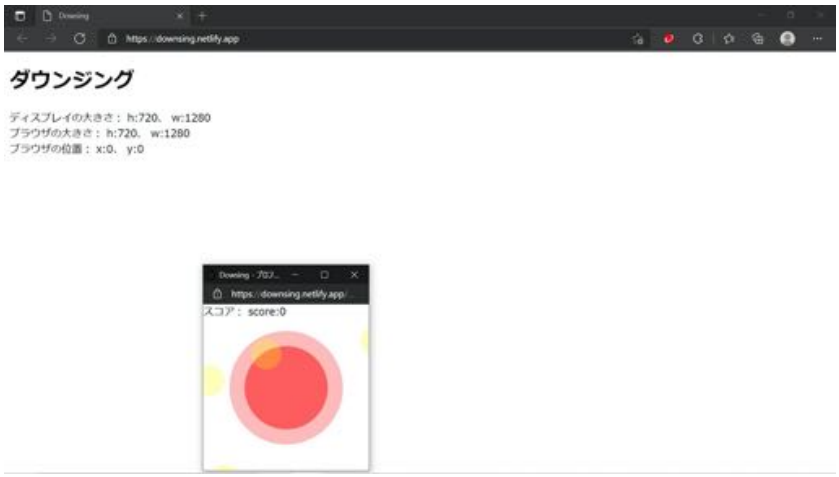
3.15.4 工夫した点

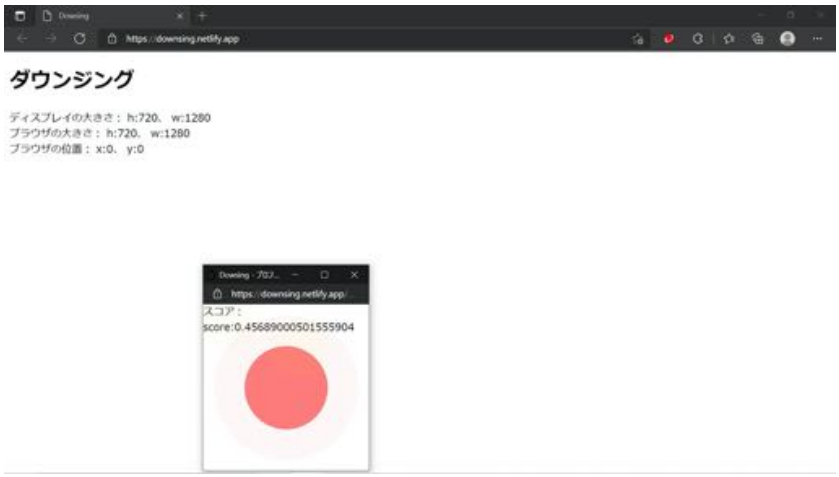
ウィンドウをレーダーに見立ててプレイするため，レーダーを表示するウィンドウの画面幅を固定したこと．埋蔵物の推理を助けるため，黄色いマーカーを設置する等してゲーム性を高めた．

3.15.5 苦労した点

画面を固定幅にする際のサイズの変更や，ディスプレイ上でのウィンドウの位置関係の計算等の計算．

3.15.6 スクリーンショット





3.16 Web ワードクラウド

文責: 星名 藍乃介

3.16.1 URL

- URL：https://ran350.github.io/web-wordcloud/
- ソースコード：https://github.com/Ran350/web-wordcloud

3.16.2 概要

文章の内容をひと目で印象付ける手法として，ワードクラウドと呼ばれる，文章中で出現頻度が高い単語を複数選び出し，その頻度に応じた大きさで図示する方法がある．今回はブラウザオンリーでワードクラウド画像を生成し表示する Web サイトを開発した．特徴は，Progressive web apps (PWA) に対応しており，イン

ストールすればオフラインでも使用できることである．ワードクラウドクラウドを生成するまでフローは，次の通りである．

1. テキストエリアから入力された文章を受け付ける
2. 文章を形態素解析して単語に分割する
3. 各形態素の出現回数を計測する
4. 一部の助詞や助動詞，特殊文字などのストップワードを除外する
5. ワードクラウドを生成する
6. 画像を表示する

このうち形態素解析する処理を Web Assembly (以下，WASM) で実行しているために，バックエンドを必要とせず，オフラインで実行することができる．具体的には，Rust の Lindera という日本語形態素解析解析器を WASM に変換し，JavaScript から利用している．

3.16.3 使用技術

- WASM
- Rust
- Next.js
- TypeScript
- GitHub Pages

3.16.4 工夫した点

ワードクラウド内の文字サイズとカラーを工夫した．文字サイズは，単語リストの出現回数に依らずおおよそ一定で，出現回数の最大値最小値によって大きくなったり小さくなったりしないよう調整した．カラーについては，グリーン系統のカラーパレットを使用し，トーンが合った出力となるようにした．

3.16.5 苦労した点

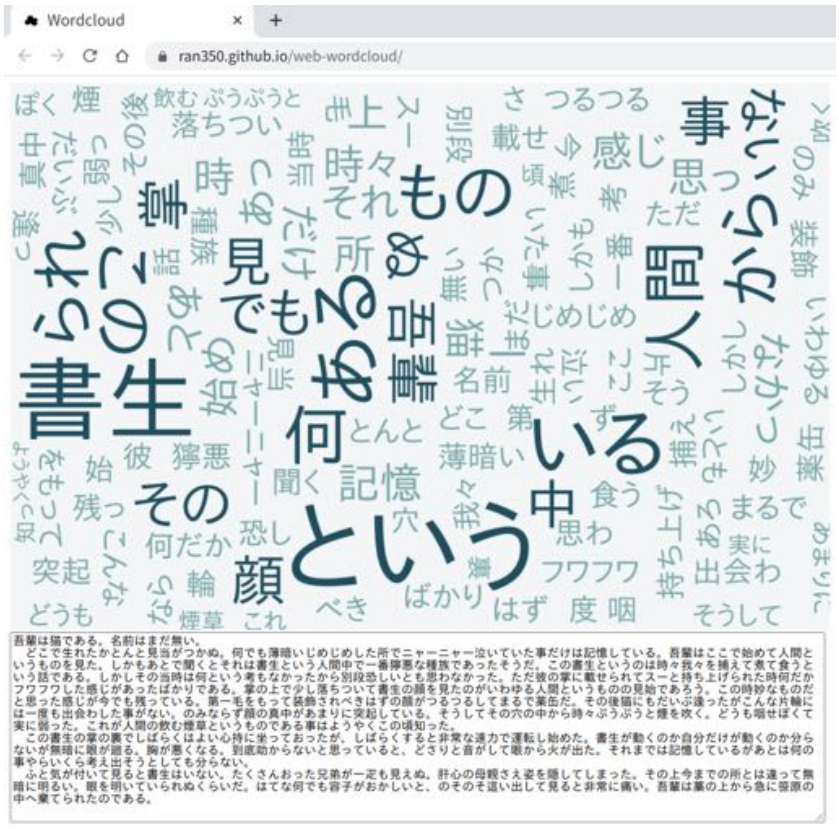
初めて Rust や WASM を使用した開発であったため，WASM ファイルを動的に読み込む処理や，Next.js と WASM 間でのデータの受け渡しの処理の実装で苦労した．

3.16.6 今後の展望

現在の UI は，ワードクラウド画像表示のための Canvas 要素とテキストエリアが配置されているのみである．今後は適切な UI 設計を行い実装したい．

3.16.7 スクリーンショット

「吾輩は猫である」の冒頭を入力した結果



3.17 onakasuita.city

文責: やぎちゃん

3.17.1 URL

- URL：https://onakasuita.city/
- ソースコード：https://github.com/ygkn/onakasuita.city

3.17.2 概要

onakasuita.city は，空腹であるという意味の「おなかすいた」という文に大阪府に存在する市である「吹田市」をかけた「おなか吹田市」という文を，「.city」トップレベルドメインを用いたタイトルと同名のドメイン名を使用した Web サイトである．

このように，インターネット上での会話やツイートの本文に使用することを目的とした，文や単語を表すドメイン名を取得する文化がインターネット上に存在する．例えば，同意を意味する「そうです」を表す「soude.su」，労いの言葉である「お疲れ」を表す「otsu.care」などが挙げられる．これらのドメイン名は，

kuso.domains [*7] という Web サイトにまとめられている．

3.17.3 開発言語

開発言語は，素朴な HTML，CSS，JavaScript のみであり，ライブラリやフレームワークは使用していない．これは，ライブラリやフレームワークのアップデートなどによるコードの変更を無くすことが目的である．

3.17.4 ドメイン

ドメインはドメイン取得・管理サービスであるバリュードメインで取得した．これは，初年度のドメイン使用料金が他サービスと比較して安価であったこと，著者に使用経験があったことなどが理由である．

3.17.5 ホスティング

ホスティングには，フルマネージドサービスである Vercel を使用した．これは，デプロイ・管理の容易さなどが理由である．

3.17.6 PWA

ユーザビリティ向上のために Web でネイティブアプリケーションのような操作感を得られるプログレッシブウェブアプリ (Progressive web apps, PWA) を採用した．

onakasuita.city の PWA 対応によって，以下に挙げる点を改善した．

- ホーム画面に追加可能
- 図 3.17.6 のようにショートカットを用いて容易に Web サイトをツイート可能
- オフラインでもページを閲覧可能

[*7] https://kuso.domains/

PWA 対応を行うにあたり，具体的に以下のことを行った．

- Web App Manifest [*8] の追加
- Create an offline fallback page [*9] を参考にした Service Worker Script [*10] の追加

3.17.7 地図の埋め込み

onakasuita.city では，吹田市の Google マップの地図をページ内に埋め込んでいる．

Google マップでは，「他のユーザーとマップやルートを共有する」ヘルプ [*11] にあるように，地図の HTML 埋め込みコードを容易に取得することが可能である．このコードは，`iframe` 要素に `width`， `height` 属性が指定されていることによる Cumulative Layout Shift (CLS) [*12] 防止や， `loading="lazy"` 属性による遅延読み込みが実装されており，ユーザビリティやパフォーマンスの観点で良い実装であると言える．しか

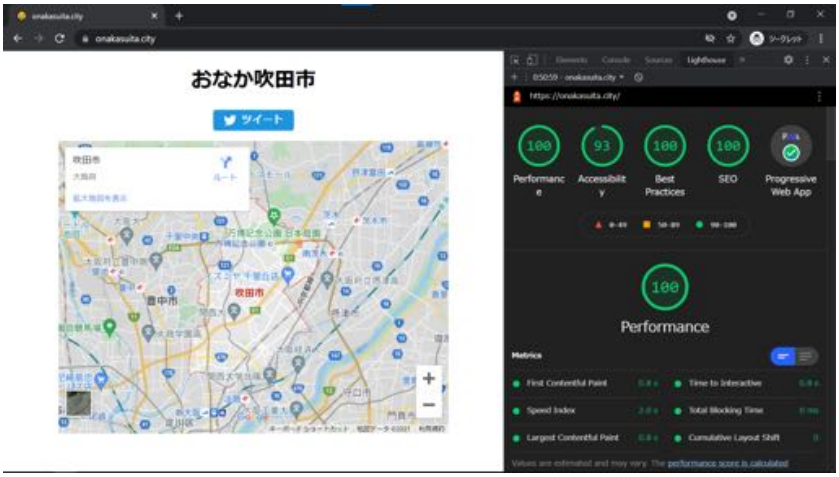
[*8] https://github.com/ygkn/onakasuita.city/blob/ea2af170438c4aacb9d61c1114d0e65f0812bcc8/manifest.webmanifest
[*9] https://web.dev/offline-fallback-page/
[*10] https://github.com/ygkn/onakasuita.city/blob/ea2af170438c4aacb9d61c1114d0e65f0812bcc8/sw.js
[*11] https://support.google.com/maps/answer/144361
[*12] https://web.dev/cls/

し，iframe タグに title 属性が付与されていないことから，アクセシビリティ上の問題[*13] [*14] が存在する．onakasuita.city では，iframe タグに新たに title 属性を付与することによりこの問題を修正した．また，スマートフォン等の横幅が狭いデバイスでページを閲覧した場合に iframe 要素がビューポートをはみ出していたが，CSS max-width: 100% を追加することにより対処した．

3.17.8 今後の展望

何か面白い機能を思いついたら実装したい．

また，現在図 3.17.8 のように，Lighthouse の Accessibility 項目で問題が発生している．これは，Twitter のブランドカラーを背景色にし，白を文字色にした「ツイート」ボタンのコントラスト比が足りておらず，視認性が低下していることによる．この問題について調査し，改善したい．



[*13] https://developer.mozilla.org/ja/docs/Web/HTML/Element/iframe#accessibility_concerns
[*14] https://www.w3.org/TR/WCAG20-TECHS/H64.html

3.17.9　スクリーンショット

図 1　PC で閲覧した場合のスクリーンショット

図 2　スマートフォンで閲覧した場合のスクリーンショット

4　総評

文責: 星名 藍乃介

制作物について，Web 開発初心者の班員も多くいたが，前章で挙げたように HTML/CSS や Canvas,React といった技術の基礎文法を習得し，実装する力を身につけることができた．教科書やドキュメントを読み進めるだけではなく実際に手を動かしながら学ぶことで，Web 開発の流れやリファレンスの参照方法，デバッグの方法などについての理解も深めることができた．

次に，反省点として，各制作物開発の進捗確認が不十分となってしまったことが挙げられる．

最後に，Scrapbox で Web 開発時に便利なツールやサービス，拡張機能，リファレンスなどをまとめていたが，本プロジェクト活動に留まらず，今後も有効活用していきたい．