Brett Tanner

bretttanner171@gmail.com My Portfolio: https://brett-tanner.dev 070-4215-9870

Full-stack Web Developer Summary

- The sole web developer for an after-school care company with over 30 schools in Tokyo/a large and ever growing number of online services
- Used Ruby on Rails to create an event registration site handling seasonal event bookings for 5 000 parents while providing management with a statistics dashboard and administrative features.
- This reduced the average time needed for a customer booking from 10-20min to 1-5min, and saved school managers hundreds of hours per seasonal event by automating aspects of the booking flow such as booking creation/confirmation emails and price adjustments in certain predefined situations.
- Created the primary information session/inquiry flows for our company, an administrative backend to manage them and a Rails API for the seasonal event site to sync such inquiries with a legacy system.
- Prospective customers now have a much smoother path to conversion which is usable on mobile and the company was able to shut down a legacy Laravel server which ran the deprecated flow.
- Currently working on a Rails-based Learning Management System to provide our curriculum to other schools in addition to handling student/school management and progress tracking. The system will also be used internally to deliver materials to KidsUP teachers, replacing a Google Drive-based system often cited as a pain point in feedback from teachers.

Skills

Tools: AWS, Brakeman, Bun, Bash/Zsh, **Docker**, **Git**, Github, Linux, **Neovim**, VSCode, Vite, WSL
Languages: CSS, Markdown, Javascript/**Typescript**, HTML, HAML, Python, **Ruby**, SQL
Frameworks: Astro, Postgres, Pundit, React, RSpec, **Ruby on Rails**, **Tailwind**
Soft Skills: Communication, **Growth mindset**, Self-sufficiency, **Problem-solving**, **Teamwork**

Experience

KidsUP **November 2022-Present**

Full-stack Web Developer

- Sole responsibility for designing, implementing and maintaining all new web projects
  - Directly interact with upper management and translate their requirements to a system satisfying those requirements in an efficient, secure and scalable way
  - Implement that system using Test Driven Development, taking into account continuous feedback and feature requests from key stakeholders
  - Style the site's frontend based on mockups provided by designers
  - Deploy test/production sites to AWS, configure various necessary AWS services such as Elastic Beanstalk, SES, Route 53, EC2 instances and Postgres RDS.
  - Upgrade AWS platforms and application dependencies as necessary to avoid security vulnerabilities
- Provide assistance with maintaining or modernising legacy systems as necessary
  - Added a new general inquiry form using plain Javascript to dynamically fetch a list of our current schools and submit inquiries to the seasonal event site's Rails API
  - Added a real-time search bar to the main site's school list page, enabling customers to filter by address, school name and pick-up/drop-off areas
  - Made various minor styling tweaks to the main site's wordpress template, such as removing a background image from the mobile hamburger menu, altering the appearance/location of text
- Promptly add features to both new and legacy systems as requested by management
  - Added a detailed statistics dashboard to the seasonal event site, showing statistics for areas like bookings, activity popularity, coupon use, information session inquiries and option popularity
  - Added survey functionality to the seasonal event site, allowing management to create surveys with a variety of question types which can be shown to students matching certain criteria

  - Allow simultaneous event and nested activity creation for all schools from a front-end interface, with activity options automatically populated from a predefined list
  - Replaced the existing iframe-based information session flow with a responsive React app
- Saved hundreds of hours of school manager time by replacing the incumbent Google Sheets/Forms system with a new seasonal registration site
  - Automated booking creation and confirmation emails, freeing school managers from writing and sending them manually
  - Automated application of common invoice adjustments like the first-timer fee, hat fee for bookings including outdoor activities and a repeater discount for external students attending a second event
  - Gave customers the ability to log in and make changes to their own bookings, preventing hundreds of requests per event for school managers to do the same
  - Switching to a system where parents are associated with their children and don't have to manually enter their information for each activity registration nearly eliminated time lost due to incorrectly entered information, once the initial association of children with parents was complete
  - Reduced the time needed to load key information like activity attendance lists and the main table showing all bookings and their details for a given event at a school from minutes to < 1 second

Projects

Seasonal Event Registration Site

- Used Ruby on Rails, vanilla CSS, Bootstrap & AWS
- Improved customer UX by replacing a repetitive Google Form based system with an account based site which provides a running total of the user's booking as it's made, invoicing, automated emails, the ability to edit bookings and the ability to view booking history
- Replaced an internal Google Sheet which took 5-10min to load with a staff interface which takes less than a second to load the same data, in addition to offering many additional features like editing bookings, confirmation emails, printing attendance sheets for activities and tracking sales statistics
- Has now been in use for 3 seasonal events and received overwhelmingly positive feedback on user experience from both customers and school managers

Information Session/Inquiry Flow

- Used React, Tailwind, Ruby on Rails & Google App Script
- Replaced an impossible to style, glitchy form embedded in an iframe with a multi-step React app styled to match the site theme and capable of being used on a mobile device
- Added a real-time search to the initial list of schools, and gave school/area managers an interface to add keywords to their school which the search will pick up
- Designed a Rails API to provide the React app with a school/information session list, and another to provide the legacy Google sheets system with inquiry data
- Built a new administrative dashboard for information sessions/inquiries into the seasonal event site
- The new flow is much more user friendly on mobile, where analytics show most people accessing our website are. The admin dashboard also allows information sessions to be created more quickly by copying details from an existing session, and managed from the same account as the event site.

Learning Management System

- Using Ruby on Rails, Tailwind, Docker, Hotwire/Stimulus & AWS
- Enabling a new business arm which is expected to provide a significant additional revenue stream
- Provides teachers with a simple view of all materials for the day upon login
- Automatic generation of PDF lesson plans from a template & information entered by curriculum staff, allowing lesson plans to easily be modernised by changing the template and regenerating all lesson plans from their data or updated through a simple web form
- Curriculum team members can propose changes to lesson plans, which can be reviewed and approved, rejected or sent back for changes with comments by an admin
- Contracted organisations can manage students/staff/schools/classes and their academic progress, as well as making support requests with image uploads within the site itself
- Will also be used internally to replace an unpopular Google Drive based system, which can be unreliable and difficult for new hires to learn