

Frank Lindsay

<http://franklindsay.me/> | C: (765) 602-8269 | frank@franklindsay.me

Summary

As a full-stack web developer, I enjoy utilizing my knowledge to code and learning new ways of doing things better. I am experienced with all stages of the development cycle from small to large-scale projects. I am well-versed in the front-end, back-end, servers, and APIs. When I don't know something, I quickly research and put it to work for my clients. I enjoy working solo as well as being a great team player. I communicate on progress and ask questions when need to make sure the project I am working on is continuously making progress.

Highlights

- Ingenious problem solver
- Passion for engineering
- Enjoyable team player
- Full stack developer
- Resourceful learning
- Excellent communicator
- Efficient project manager
- Creative design patterns
- Clean documented coder
- Strong decision maker

Experience

Rebel Rates

<https://rebelrates.com>

- The original codebase was procedural PHP and was rebuilt using the Laravel 8 framework.
- To manage the data, content, and users Laravel Voyager was integrated to give administrators control over the platform.
- Stripe was utilized to charge the users for subscriptions and donations made on the system.
- An affiliate system was built with Stripe promo codes/coupons generated from Laravel Voyager to help track referrals and reduce subscription charges.

- Utilized a user role system to limit parts of the system for administrators, affiliates, agents and IMOs.
- A dynamic dashboard was built for agents that gave them complete control over their profiles. This includes subscription management, image uploads, product availability, custom URLs, contact information, and services offered.
- The affiliate and IMO accounts have a dashboard to manage referrals which includes searching, exporting and customizing reports.
- Mailchimp was used to send mail for all aspects of notifications. This included registration, administration notifications, affiliate notifications, agent content and automated emails.

Automate.Ink

<https://app.automate.ink>

- Automate is an application built using the Laravel 8 framework.
- Guzzle library was utilized to communicate with the Printavo API to pull in data.
- The gateway design pattern was used to help divide logic of code up to stick with the DRY principle and keeping the codebase clean.
- The Cashier library that utilizes the Stripe payment system was used to handle the subscription payments.
- A profile was designed to let users update their profile, invite team members, enter the Printavo API token settings and manage billing.
- Building reports allows users to define types (csv or live widget), select collection of data to use, choose headers to use and specify filters to pinpoint data to utilize in reports.
- Apex Charts is utilized to build dynamic graphs based on the data provided by filters selected by users.
- The chart widgets can be used to company current and previous date periods.
- Selecting date periods can be predetermined such as last week, last month, last year, etc as well as custom date periods are allowed to be entered.
- Widgets can be shared with team members, deleted, csv file generated from widgets and reports can be downloaded.
- Orders can be searched on the system that exist on the Printavo API system.

Campus.Ink

<https://vault.mycampus.ink>

- The application originally started as a procedural application built in PHP and rebuilt with the Laravel framework.
- The Guzzle library is utilized to communicate with the Printavo API to pull data.
- Utilize services such as Github, Codeship and Linear to manage delivery of code to the server.
- Quotes and Orders have integrated filtering to handle finding specific data based on statuses, dates and users.
- Used design patterns such as gateways to organize code and stick to the DRY principles.
- Calculations were used for commission, bonus, fundraiser and greek licensing.

- A simple payment system to mark orders as commission payed was integrated to help the company administration help keep track of the books.
- A projects system was integrated for students to help create new orders with qualifying questions, customer details, quotes and notes feature to help the student process.

M&C Assemblies

<https://assemblytoolbox.com>

- Built a contract platform that allowed generated contracts and allowed signature to be inputted with PDF documents.
- Updated the Laravel framework to new version.
- Worked with a team of engineers to collaborate on updating and adding new features.
- The initial build utilized MySQL and migrated to MongoDB for data storage.
- Integrated jQuery throughout the site to make a more responsive and communicate with the back-end server without refreshing the page.
- Developed many functionalities that helped manage jobs and technicians.
- Bootstrap was used to help make the site mobile friendly.
- Utilized pem and ppk keys for communicating with third party servers securely through encryption.
- Worked with AWS platform for storing files, developing and server management.

CTX

<https://www.ctxfacilityservices.com/>

- The website was created using the Laravel framework for future growth.
- The repository of choice used was Github for storing code.
- The site was designed to be mobile-friendly as well as look good on a desktop.
- Forms utilized allowed potential clients to message the company that would be stored in the database as well as being sent out as an email for quicker response.

School Calculators

<https://schoolcalculators.com>

- This was a project that was already built with the Laravel framework but needed some refactoring.
- Needed to create a new AWS server along with new security keys for secure access.
- Fixed the Stripe integration that had a few bugs from successfully charging a credit card.
- Integrated a discount system for multiple purchases of calculators.
- The mailing system was not functioning correctly and needed to fix bugs.

Qwire

<https://www.qwire.com/qwireclear>

- Qwire Clear was a project that was built on Laravel 5 that was already in

- progress.
- The project required VueJS for the front-end for the UI and to communicate with the back-end server.
 - Azure was utilized for authentication so that organizations could use their secure accounts with the platform.
 - Gulp and elixir were used to handle the CSS and javascript for the front end.
 - HTML and CSS were heavily used to create a sleek management dashboard for users.
 - GitHub with peer reviews and testing was utilized for bug-free and efficient code.
 - Project management used Agile development with the Jira platform.
 - There were several bugs throughout the system that I was responsible for fixing.

My Student's List

<https://mystudentslist.com>

- Utilized VueJS and the Laravel framework to build an e-commerce single-page application (SPA).
- Built an administration panel that managed invoices, content management system, and many other elements of the application.
- Integrated Octobat, Shippo, and Stripe APIs to manage payments, shipping, and tax of user sales.
- Used Twitter Bootstrap framework to build both a desktop and mobile user interface.
- Completed all work alone and with very little direction to devise a solution for the requirements with satisfactory results.
- Built a server to host and deliver the platform on a LEMP stack using DigitalOcean.

GroupVitals

<http://www.groupvitals.com>

- Worked with a small team to design and develop a customized web application for managing groups with the PHP Laravel framework.
- Built a migration system to migrate data from the GroupForm's application.
- Helped integrate the Stripe subscription service system based off number of groups a church had.
- Utilized Mandrill to send system messages and for allowing users to send customized messages to their base users.
- Helped and eventually took over the Fellowship One API integrations which involved pulling and pushing data on all of their available APIs.
- Built a queueing system that used both Supervisor and Beanstalkd that handled background services such as sending email and other jobs of the backend.

- Managed the cloud servers such as dotCloud and DigitalOcean that utilized Nginx.
- Integrated jQuery to handle much of the JavaScript work such as event calls and ajax.
- Started integrating PHP Unit testing to help reduce bugs in the production deployments.
- I took on the responsibility of having the project utilize continuous integration with Git, Codeship, and code reviews for faster and optimized development.
- Built custom backup solutions that used AWS S3 for storage.
- Took part in integrating multiple APIs for analytics such as Segment, Intercom, and Google Analytics.
- Once the team was reduced, I took on the responsibility of helping with support for the users.
- I set up systems to help alert when errors and failures were occurring in the backend to quickly fix bugs.
- Integrated Homestead Vagrant into our deployment to quickly allow anyone to spin up a server for the project locally.
- Refactored much of the code to use design patterns such as gateways, repositories, factories, and many others to make the code cleaner and much easier to test using dependency injection.

GroupForms

- Designed backend code and database to use with the PHP CodeIgniter framework for group management software.
- Used the AWS S3 system for saving files for the users and storing backups of the system.
- Integrated APIs such as Segment for analytics and Mandrill for sending emails.
- Utilized jQuery library for cross-browser ajax and other features such as column drag and drop sorting.
- Integrated a migration system to export data to the GroupVitals system.
- Built a form system that users could customize to share via a link for their prospects.
- Managed the servers on cloud systems such as DigitalOcean.
- Later started doing Git usage and code reviews with other developers.

GetHuddle

- Find and fix bugs that were littered around the system.
- Improve the user interface design with Twitter bootstrap.
- Move the system from a cPanel type server to a customized cloud based setup.
- Integrate a huddle (group) creation limitation feature.
- Optimized mailing system to fix image sending and prevent emails from enter the spam/junk folder.

Fancy Hair

<http://www.getfancyhair.com/>

- Worked with the PHP CodeIgniter framework to develop an ecommerce platform from scratch.
- Carried out quality assurance tests to discover errors and optimize usability.
- Integrated multiple APIs such as SmartyStreets and MaxMind.
- Customized PayPal open source libraries to allow the use of the IPN.
- Developed an entire administration panel to control the website and orders.
- Added an affiliate system to track referral of customers from users.
- Built a gift card system that created cards and added to the PayPal IPN.
- Later developed a blog feature that allowed text and video entries based on categories.
- Eventually gutted the original design and replaced it with a much sleeker look.
- Created multiple CSV export features for orders, affiliates and accounting.

CMSIgnited

- Utilized CodeIgniter framework to build custom content management system.
- Created an administration panel to control both the front-end and back-end of the system.
- Integrated open source libraries that enabled module based features.
- Built email templates, files, pages, variables and widgets modules that allowed the management of content throughout the website.
- Developed the navigation and theme modules to allow the control of the design of the front-end.
- For the member management, I built a groups, members and permissions module that allowed not only the authentication of users but manage the areas that those users had access too.

- Redirects and settings made up the tools of the system to allow the control of flow throughout the website.

Bust-A-Code

- Utilized PHP and MySQL to develop a full freelance website.
- Integrated PayPal API for payment system.
- Created web projects management that allowed the posting, bidding and choosing over developer.
- Built web scripts management that allowed the posting, searching and download of software.
- Designed project posting to match the details to web scripts at the end of the posting. This allowed clients to invite developers that have already created projects similar to theirs. Once invited, the developer can bid less than the set amount, as they can use their script to build the client's project.
- Developed profiles that allowed the management of bids, saved projects, financials, resume, portfolio and my scripts.
- The administration panel allowed the management user's, projects, scripts, categories and any other item part.

MyToDoList

- Created a todo list management system with PHP and MySQL.
- Built an authentication system that allowed registration, account activation and login.
- Developed the todo system that allowed CRUD operations and the assigning of the todos to other users.
- I also created tabbed categories for the todos. This allowed a user to view theirs, others and closed todos.
- I also utilized ajax to retrieve most of the data and updating.

LI2C (ver. 2)

- A MLM that was developed using PHP and MySQL
- Developed an infinity wide 2-tier affiliate backend system.
- Created authentication system that allowed access to their profiles, referrals and manage their commissions.

- Built an administration panel that allowed administrators complete control over the front-end.
- The administration panel included features such as quick statistics, member management, file uploads, financial management, CMS tools.
- I also created resource managements such as advertising resources, testimonials, recommendations, banners and email advertising.

LI2C (ver. 1)

- The first version of the LI2C system that I developed using PHP and MySQL.
- Created a 4x4 matrix system affiliate system.
- Developed an authentication system that allowed users to register under a sponsor, login and forgot password.
- Profile, statistics, messaging system, product access and promotional tools were developed to help users manage their referrals, leads and account.
- I created the administration panel to help manage the ecosystem. This allowed the management of members, products, uploads and CMS.

Programming Languages

HTML, CSS, JavaScript, Twitter Bootstrap, jQuery, VueJS, PHP, SQL, CodeIgniter, Laravel, Stripe, Multiple APIs, Design Patterns, Beanstalk, Supervisor, Guzzle, GIT, Continuous Deployment

Education

Highland Highschool

2000-2004
Anderson, Indiana