Ashley Phillips
Email: ashp1621@gmail.com
Mobile: +44 7783 988250
Website: www.ashphillips.com

PROFILE

SUMMARY
Experience in big data real-time analytics, many different kinds of applications from banking to e-commerce in several programming languages and frameworks across the full stack and software life cycle, with an understanding of the need to balance time constraints with careful design and best practices.

EDUCATION
University of Brighton
2011 – 2014
BSc Mathematics 2:1

SKILLS

KEY SKILLS
<u>Full-stack:</u> JavaScript, WebSockets
<u>Front-end</u>: React, React Query, Redux, HTML/CSS
<u>Back-end</u>: Lambda, Python, Node.js, C++, Express, MongoDB, PostgreSQL, SQL, NoSQL, Database Design
<u>Testing</u>: TDD, BDD, Mocha, Cucumber, Jasmine, Chai, Selenium, Cypress
<u>DevOps</u>: Docker, AWS, Bash, Webpack & Babel, Git, NPM

OTHER
Java, nginx, Qt, Documenting APIs, Jira & Scrum, Software design patterns & algorithm design, Mathematical modelling, statistical analysis, French (B2+)

PAST EMPLOYMENT

January 2024 – June 2024: Student of French (B2+)
Obtained B2 – C2 in the TCF Canada exam.

Senior Software Developer, Plentific Ltd, London UK
February 2022 – January 2024
Frontend: TypeScript, React, React Query, Alt.js (Flux)

Worked on a marketplace-like web app, part of a vast microservice architecture. I owned several projects from conception to release, and worked as part of both large and small teams. I also garnered a reputation for refactoring large chunks of *hellish* code. I made heavy use of React Query here after it was introduced to replace Flux, which simplified data flow, while reducing unnecessary data fetching.

Senior Software Developer, Feedr Ltd, London UK
January 2021 – February 2022
Full-stack: JS/TypeScript, React, MongoDB, Node.js, iOS & Android

Worked with various services, web apps and mobile apps to connect food suppliers with employers. Duties involved mentoring junior devs, owning projects, unit testing, feature development/enhancement, code reviews.

Software developer, iManage, London UK
June 2019 – June 2020
Frontend & infrastructure: JS, Docker, React, Redux, Ramda, Cypress, Styled Components

Worked on the frontend, and devops. Created/improved features to allow users to extract data from potentially millions of documents via AI. I also created a custom solution to run cypress tests in parallel, which massively sped up completion time of the test suite.

Frontend developer, SuperCarers, London UK
March 2018 – June 2019
Frontend: JavaScript, Python, React, Redux, Styled Components, Jekyll, Selenium, Jest, Jasmine, Enzyme, Chai

I worked on both the website and the app as well as sharing the diagnostis and fixing live issues.

Software developer, Dev2Rights, London UK
July 2017 – November 2017
Full-stack: JavaScript, Node.js, React, Flux, AWS (Lambda, EC2, API Gateway, S3, RDB, CloudFront, CloudWatch, IAM, JavaScript SDK), PostgreSQL

A development agency working on a recruitment app. We made heavy use of AWSLambda, S3, CloudFront, React & Flux.

Software Developer at Ancoa, London UK
Feb 2015 – April 2016
Full-stack: C++, Qt framework, JavaScript, HTML & CSS, Dart & AngularDart, Linux, Git, PostgreSQL

I created algorithms to help detect fraudulent behaviour in stock markets, as well as contributing to other areas of the system, data-integration, database management, linux build system, desktop and web front-end, client APIs and some client-facing tasks and attendance of Fintech events. Joined the web team full-time for last 5 months there and helped develop the web interface.

Graduate Developer at Sword APAK, Bristol UK
July 2014 – Feb 2015
Full-stack: C++ (.NET), SQL Server

Enhanced and developed new features, while working closely with business analysts.

PERSONAL PROJECTS

Unreal Engine 5 (2023 – Present)
C++, BluePrints, UE5 API

A hobby of mine, I have prototyped characters and a combat system, hopefully to be part of something larger in the distant future.

Spaced repetition app (Sep – Dec 2020)
JS, Docker, React, Redux, MongoDB, Mongoose, Node.js, Database Design

Personal project to create a [spaced repetition](#) app, as I was unhappy with what was already out there while learning Japanese as a hobby ([Memrise](#) and [Anki](#) being the main ones at the time).

Client-server Room package (2016/17)
ES6, Node, WebSockets

The Room package is designed to be used whenever there are multiple clients communicating with each other via a server, with otherwise complex functionality made much easier to implement.

Samples, git repo, tutorial & API Docs:
Server: https://github.com/Nodeocrat/server-room
Client: https://github.com/Nodeocrat/client-room
Samples: https://github.com/Nodeocrat/room-samples
API Docs: https://github.com/Nodeocrat/room-samples/blob/master/api-docs.md

Lobby/Chat App (2016/17)
ES6, Node, React, Redux, Socket.io, Express.js, CSS3, HTML5

A lobby with a chat room which has been integrated with the loop-less game server project mentioned below. Players can choose to create new instances of this game, or join one. This project inspired me to write the Room package, and has itself been re-written to use the Room package.

Loopless game-server (2016/17)
ES6, Socket.IO, Node.js, HTML Canvas

Multiplayer game experiment with JS and Node for the backend with Socket.IO. Built using JavaScript with no other third party libraries; The back-end runs on a loop-less game-server and is completely event driven, with collisions and positions only being calculated as and when they need to, rather than 30 times per seconds (as is usually the case), which greatly reduces server load. Multiple players can log in and shoot at each other and shrink/enlarge in size. Socket.IO has since been abandoned in favour of regular WebSockets (provided by Room package).

Personal website (2016/17)

ES6, React, Redux, Handlebars, AWS, Linux, Node.js, Express, MongoDB, Mongoose, passport (local, google, facebook strategies), Socket.io, bcrypt, Nginx

Website has had 3 re-writes. First version used vanilla JS with handlebars. Second version was written using React with the third version with React + Redux. Website is not currently hosted.

Auslander-Parter Algorithm (2014/15)
Java, Algorithm design

Software which demonstrates various Graph Theory algorithms written in Java, with the option to execute algorithms on custom drawn graphs.
https://github.com/Nodeocrat/Graph-Theory-Algorithms/blob/master/src/Algorithms/AuslanderParter.java