|  |  |  |
| --- | --- | --- |
| [dhruvs7@outlook.com](mailto:dhruvs7@outlook.com)  Rutherford, NJ. | **Dhruv N. Shah**  <https://shahd5.github.io/> | 201-774 - 2768 |

**Objective**

Seeking the position of a Software Engineer to further enhance organizational worth owing to my knowledge in Object oriented programming, and Data Structure.

**Education**

**Bachelor in Computer Science Engineering May 2018**

*State University of New York at New Paltz*, New Paltz, NY

* GPA: 3.05

Relevant Course: Object Oriented Programming, Programming and Data Structure, Foundation in Computer & Data Science, Software Engineering, Web Programing.

**Skills**

* **Language:** Java, JavaScript, Typescript, C#.
* **Frameworks:** JavaFX, Spring, Ionic, Angular 4, Node.js, Android SDK, ReactJs.
* **ORM:** JDBC, Node ORM 2.
* **Database:** MySQL, MongoDB, Google Fusion Table.
* **Tools:** Android SDK, GIT, Maven, Tomcat, Visual Studio Code, Eclipse, Linux, Microsoft , Mac OS x.

**Experience**

**State University of New York– Android Developer (Internship) January 2018 – May 2018.**

* A group of six members made the application which helped the users to locate and connect them to nearby service providers. The users will be able to use filters to narrow down their search for education, finance, housing etc.
* Front-end was developed in XML and logic in Java. A CURD service made calls to web-API hosted locally.
* Backed-end Rest API, was developed in Node.js where activity like deep-linking, filter and sorting of data was managed. A CURD service loosely coupled with cloud hosted Google Fusion table database
* Testing of web-API was done with mocha and chai. Front-end testing with espresso.
* **IDE** - Android Studio | **Framework** - express.js, express-deep-link | **Testing** - mocha, chai, JUnit | **Language** - Java, XML, Node.js | **Database** - Google Fusion table.

**State University of New York- Full Stack Web Developer May 2016 – September 2017.**

* A web-application was designed for the faculty of SUNY to make request to purchase material for their research.
* Front-end was developed in Angular 2. A service component made Ajax calls to Rest web-API.
* Backed-end Rest API, was developed in Node.js where activity like deep-linking, filter and sorting of data was managed. A CURD service loosely coupled with Microsoft SQL database.
* Testing of web-API was done with mocha and chai. Front-end testing with Karma.
* **IDE** - Visual Studio Code **|** **Language** – node.js, typescript **| Framework** - express.js, express-deep-link, Angular 2 **Testing** - mocha, chai, Karma. **| Database** - Microsoft SQL

**Academic Projects**

**Payroll application – Full Stack Web-Application System (SUNY Academic Project) July 2017 – Jan. 2018.**

* A user can log in a web-based application to clock in and out from work, check their payroll, and other activities.
* Front-end was developed in HTML and logic in JavaScript. jQuery library made the Ajax calls to web-API.
* Backed-end was developed in Node.js. A service provider tightly coupled with MySQL database.
* **IDE** - Visual Studio Code | **Language** – Node.js, JavaScript | **Library** - jQuery | **Database** - MySQL

**Fitness Website** **– Full Stack Web-Application (SUNY Academic Project) July 2017 – Jan.2018.**

* A fitness website that tracks users progress towards their fitness goals and suggests them diet and exercise.
* Front-end was developed in Angular 2. A service component made Ajax calls to web-API.
* Backed-end API was developed in Node.js where activity like routing and POST static data was managed.
* **IDE** - Visual Studio Code **|** **Language** – node.js, typescript **| Framework** - express.js, Angular 2.

**Personal Projects**

**Label Reader Application – Cross-Platform Mobile Application May 2018 – Present.**

* Cross platform mobile app that read ingredients from food packages and notify the users with allergy/toxic ingredients.
* Front-end was developed in Ionic. A CURD service makes calls to web-API.
* Backed-end Rest API, is developed in Spring where activity like deep-linking, filter and sorting of data is managed. A loosely coupled CURD service connects to Google cloud platform and Microsoft SQL.
* **IDE** - Visual Studio Code, Eclipse | **Framework**- Ionic, Spring-boot | **Language** - Java, typescript | **Database** – Microsoft SQL.