|  |  |  |
| --- | --- | --- |
| **Git Hub:**<https://github.com/nilesh507> | **Nilesh Bhoi** | [nilesh507@gmail.co](mailto:nilesh507@gmail.com)m |
| **LinkedIn:**[www.linkedin.com/in/nileshbhoi/](https://www.linkedin.com/in/nileshbhoi/) | | 315 4365644 |

**Academic Performance**

|  |  |  |  |
| --- | --- | --- | --- |
| **Institution** | **Degree** | **GPA** | **Graduation Data** |
| Syracuse University | (MS) Computer and Information Science | 3.4 | May 2024 |
| Indraprastha University | (BS) Information Technology | 77% | August 2021 |

**Skills**

**Programming Languages:** Python, Java, JavaScript, Haskell, C/C++.

**Programming Skills:** Bootstrap, jQuery, Ajax, Node.js, Express.js, SQL, Oracle, MongoDB, Layouts & Partials, Passport(JWT, oauth2), CSS, SASS, Multer, NodeMailer, Socket.IO, React, Firebase, Redux, NumPy, Pandas, Matplotlib, RESTapi.

**Tools:** Linux, Eclipse, IntelliJ, VS Code, Git, WebStorm, PyCharm, Postman, Robo 3T, Firebase, Jupyter notebook, Spyder.

**Experience**

**Teaching Assistant,** Coding Ninjas, Delhi, IndiaMarch 2020 - July 2020 **(Java)**

* Mentored people across India, monitored students’ performance and cleared their doubts about Data Structure and Algorithms.
* Total number of Doubts taken: 600 plus with rating: 4.7/5

**Coursework and related Projects**

* **OPERATING SYSTEM (C/C++)**
  + Read multiple user programs and schedule them preemptively and non-preemptively.
  + Make **Unix system calls** - read, write, yield, exit.
  + Imitated RAM with the virtual address to physical address translation management in NACHOS.
* Built a ready queue and scheduler effectuated with the timer interrupt service routine in a **multi-programming** environment.
* Designed multiple scheduling algorithms for processes/threads with variable time quantum for performance comparison.
* **Structure Programming And Formal Methods (Haskell)**
* Built a property testing model for random/automatic **property testing** for the Haskell Programs.
* Plot a deep **automated theorem proofer** with the goal to express access control policies and their ramifications.
* Given access control matrix determined the accessibility of the requests made.
* Bell-La Padula security and Biba strict Integrity model were used to determine the grant of requests.
* [**Major Project: Steganography**](https://ijtre.com/wp-content/uploads/2021/09/2021081105.pdf)  **(Java)**
* Encrypted a secret message within an image with a secure approach that uses spatial domain image steganography.
* It includes a **Least Significant Bit (LSB) algorithm** for the encryption built in java.
* The cover image and encryption information converged to a stream of bits effectuated with BitStream and vice versa.
* For encryption and decryption-cover image was broken up into individual bit-plane each embedded with different levels of information with a **lossless compression** technique.
* [**Minor Project: Bitcoin Price P**](https://ijtre.com/wp-content/uploads/2021/10/2020080320.pdf)**rediction (Python)**
* Predicted the price of Cryptocurrencies with Deep Learning using Bitcoin to provide insights into future bitcoin trends.
* Analyzed the time series of bitcoin prices with Autoregressive–moving-average, Autoregressive Integrated-**moving average**, and **Recurrent Neural Network**.
* [**Social Media Website**](https://github.com/nilesh507/Nilesh_Bhoi) **(JavaScript)**
* A fully responsive website following MVC architecture pattern secured with three layers of security checks with PassportJS.
* Pages are dynamically created (layouts and views) and the Website's API is secured with JWT Authentication.
* Users can post their photos along with captions. Features like, comments, and follow.
* Make friends with other users alongside a ChatBox-chat built with the help of Socket.io.
* [**Movie-react-app**](https://github.com/nilesh507/Movie-App) **(React)**
* Designed a Movie review app where you can search for movies, and add/remove them from our favorites.
* Implemented a website using **RESTFUL API**-OMDB- to retrieve the states of the movie searched.
* **Cart-react-app** **(React)**
* Designed a cart module with randomized items with **CRUD functionality** appended to them on the **firebase database**.
* App follows component update lifecycle to update the state of items, hosted on firebase.

**Training**

* [**Advance Web Development with React**](https://ninjasfiles.s3.amazonaws.com/certificate1039637c17f87f8e8eb78efd110912dbc4d4090.pdf)[4 months]
* [**Full Stack Front Web development with Node.js**](https://ninjasfiles.s3.amazonaws.com/certificate719289a0041f840f982eee2e76cd4b0ea95902.pdf)[4 months]
* [**Front End Web Development with Node.js**](https://ninjasfiles.s3.amazonaws.com/certificate719290dba9980b22b51305f2f3665a6706d5bb.pdf)[4 months]
* [**Java Foundation with Data Structures & Algorithms**](https://ninjasfiles.s3.amazonaws.com/certificate37598043dc02253543b43c307491d151258de8.pdf)[3 months]