

# Piyush

pshroy@gmail.com | +91 8102447871

## EDUCATION

### JSS ACADEMY OF TECHNICAL EDUCATION, NOIDA

B.TECH (INFORMATION TECHNOLOGY)  
JULY 2019 - JULY 2023 | Noida  
CGPA: 7.6

### DELHI PUBLIC SCHOOL AISSCE

2018 | Patna, India  
91%

### DELHI PUBLIC SCHOOL AISSE

2016 | Patna, India  
CGPA: 10

## SOCIAL LINKS

LinkedIn:// piyush0406  
GitHub:// piyush0406

## COURSEWORK

### UNDERGRADUATE

- DBMS
- Web Technology
- OOPS
- DSA

## SKILLS

### HANDS-ON

- Javascript
- Python
- React.js
- Node.js
- Express.js
- MongoDB
- Mongoose
- Docker
- Git

## EXPERIENCE

### KAZAM | BANGALORE | SOFTWARE DEVELOPER

JANUARY 2023 - PRESENT

- Implemented an **Energy Management System (EMS)** to monitor and optimize energy usage across large-scale facilities.
- Configured **Single Line Diagram (SLD)** integration with smart meter mapping and visual previews for seamless energy tracking, enhanced UX through drag-and-drop components.
- Developed a centralized **EMS Dashboard** with KPIs like energy consumption, smart meter uptime, and charger efficiency.
- Integrated IoT-enabled **data loggers** and smart meters for real-time energy data collection and system insights.
- Delivered advanced insights into **electric bus charging** behaviors, energy use, and battery management.
- Identified depot inefficiencies through detailed **energy loss reports** comparing grid imports, distribution, and utilization..
- Enhanced electricity billing accuracy with **dynamic tariff calculations**, peak demand penalties, and off-peak rebates.

### MHPL | DELHI | SOFTWARE DEVELOPER INTERN

JUNE 2022 - SEPTEMBER 2022

- Worked as a **Frontend Developer** to design and develop company's **UI/UX**.
- Translated Design and Wireframes into high quality code using **ReactJS**.
- Deploying, managing, and developing **MongoDB clusters**.
- Performed **CRUD** operations on the data imported.
- Created custom **APIs** to fulfil application requirements using **NodeJS**.

### URGESTURE | NOIDA | FULLSTACK DEVELOPER INTERN

OCTOBER 2021 - MARCH 2022

- Developed a launch website for the client's company that boosted their **revenues by 50%**.
- Application development was based on **LAMP** stack.
- Implemented a **Fullstack Web Application** that allowed NGOs to crowdfund.
- Successfully established dashboard for 3 types of users – **Donor, NGO, and Admin**.
- Hands-on experience in deployment using **Git** in **AWS** (Route53, EC2)

## PROJECTS

### SECURITY PRIORITIZATION BASED ON VULNERABILITY DESCRIPTION

- Developed an **automated vulnerability classification** system using deep learning techniques, including state-of-the-art models such as **BERT** and **RoBERTa**, to enhance cyber threat intelligence efforts.
- Evaluated and demonstrated the effectiveness of the system using **2,72,385** vulnerability reports, showcasing superior performance compared to traditional methods and even outperforming advanced models like **SVM, CNN-LSTM**, and **graph convolutional neural networks**.
- Employed BERT and RoBERTa **tokenizers** to **preprocess and encode** textual data, ensuring compatibility with the corresponding models for vulnerability classification. This facilitated effective utilization of contextual embeddings and improved the accuracy of **vulnerability categorization** within the automated system.
- Improved the accuracy and efficiency of security products by providing more accurate and time-saving validation of vulnerability data.

### QUIZMASTER

- Build a simple quiz web application in which you can play a quiz and add new categories and new questions.
- For Frontend part **React.js** is used and in Backend **Node.js** in which all **CRUD** operations are performed and for **Database NoSQL(MongoDB)** and both Frontend and Backend are deployed separately and interact through API.
- Build **Login, Signup, User Confirmation** and **OTP Confirmation (through NodeMailer)** functionalities from Scratch.

### ONLINE IDE

- Built an **online IDE** for running **JAVA , C++,PYTHON** programs.
- For **frontend**, **ReactJS** was used, for **backend**, **NodeJS** was used and **child-process exec()** was used to execute **shell commands** for compiling codes.
- Docker** was used to make necessary compilers available to the backend.