HIG-3, Pink Flats,
1 Circular Road, Rajapur
Prayagraj, 211001

Rahul Chhabra

+91-8319591025
iit2021096@iiita.ac.in
rahul29112002@gmail.com

Github LinkedIn

Portfolio Blog

Education

- **Undergraduate** : *December 2021 - June 2025*
  - Bachelor of Technology in **Information Technology** from **Indian Institute of Information Technology**
  - CGPA : **7.98 / 10**
  - Courses : Operating Systems, Computer Networks, Database Management Systems, Object Oriented Methods, Automata Theory
- **Indian Schools Certificate Examination,** *2021* 85%
- **Indian Certificate of Secondary Education,** *2019* 92.67%

Software Projects

- **Semester Project for Software Engineering Course**
  - Co-authored the backend using **Spring Boot** and **Kotlin**.
  - Developed a **RESTful API** that supports **CRUD** operations for the entire database schema, including users, publications, resources, and more.
  - Implemented an object-relational mapping (**ORM**) layer between the **H2 database instance** and the server application using **Spring Data JPA**.
  - Utilized a **Model View Controller (MVC)** architecture for **server-side rendering**, populating template variables in **Mustache templates**.
- **k8 - A CHIP 8 Emulator**
  - Developed a **portable CHIP-8 emulator** in Kotlin with support for **JavaFX**, enabling seamless execution across platforms.
  - Implemented a **modular CPU architecture**, facilitating easy integration with various graphics interfaces for diverse frontends.
  - Employed **message passing synchronization** between the drawing coroutine and CPU coroutine, resulting in impressive performance with a **peak FPS of 133** on the JavaFX frontend.
- **A Scheme to JavaScript Compiler**
  - Developing a **compiler** that converts a subset of the Scheme programming language into executable JavaScript.
  - Utilising a slight variant of the **untyped λ calculus** as an intermediate representation in the compilation process.
  - Leveraging the λ calculus representation to **generate JavaScript code**.
- **A Scheme interpreter written in Scheme**
  - Developed an **interpreter** for the Scheme programming language written in Scheme.
  - Implemented the interpreter as a series of interpreters for **successively more complicated subsets of Scheme**.
  - The design of the interpreter was influenced by Friedmann's book (**"Essentials of Programming Languages"**).

Research Project

Formalising the coinductive trie representation of regular languages in Cubical Agda *June 2023 - Present*

- Researching the application of **cubical type theory** to program and prove the completeness of a **coinductive trie** representation.
- Referencing Traytel's method [Traytel 2016] of viewing **regular languages as coinductive tries** and the proof of their representation's completeness.
- Highlighting cubical type theory as a powerful type system capable of encoding mathematical constraints and enabling **"propositions as types"** concept. [Wadler 2015]

Technical Writing

- **The Dark Side of the Nullable Moon**.
  - Authored a technical article explaining the **nullable type constructor** in Kotlin.
  - Provided detailed insights into the **JVM level representation** of nullability in Kotlin.
  - Explored the interaction of nullability at the type level with **inheritance** and **subtyping** in Kotlin.
- **Using fixedpoint combinators to implement recursion**
  - Authored an article discussing hurdles encountered while developing a Scheme interpreter.
  - Explored challenges in **implementing recursion** within the interpreter.
  - Demonstrated the use of **fixed-point combinators** as a solution for seamless recursion implementation.

Skills

- **Languages:** Java, Kotlin, C++, Rust, Scheme, Haskell, Agda
- **Frameworks:** JavaEE,Spring Boot, Hibernate
- **Tools:** Shell, Git, Github, Pandoc, Gradle
- **OS:** Linux