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

- **BTech Information Technology** *Dec 2021 - June 2025*
  - CGPA : 7.98 / 10
  - Courses : Operating Systems, Computer Networks, DBMS, Object Oriented Methods, Automata Theory
  - Extracurricular Activities : Senior Member, Music Society. Coordinator, Literary Society.
- **Intermediate** *May 2019 - April 2021*
  - 12th grade, ISC 85%
  - Participated in Google Code-In 2019-20.
- **High School** *April 2008 - April 2019*
  - 10th grade, ICSE 92.67%
  - Participated in Google Code-In 2017-18.

Software Projects

- **IIIT A Software Engineering Research Lab Website**
  - Developed a RESTful API that supports CRUD operations for users, publications, learning resources and research scholars for normal and administrative users.
  - Implemented an ORM layer between the H2 database and the server application using Spring Data JPA.
  - Performed server-side rendering (instead of client-side rendering) to optimise for infrequent data changes.
  - Tech : MVC Architecture, Spring Boot and Kotlin.
- **k8 - A CHIP 8 Emulator**
  - Implemented the fetch-decode-execute cycle of the CHIP 8 CPU architecture.
  - Implemented the graphics context interface using JavaFX, enabling the CPU emulator to be completely decoupled from the graphics library.
  - Exploited atomic booleans to implement interrupt handling between the graphics and CPU coroutines.
  - Achieved 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 variant of the untyped λ calculus as an intermediate representation in the compilation process.
- **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

- Researching the application of cubical type theory to program and prove the completeness of a coinductive trie representation.
- Exploring the application of cubical transport to automatically convert programs written in the trie representation to the set-theoretic representation.
- Highlighting cubical type theory as a powerful type system capable of encoding mathematical constraints and enabling ”propositions as types” concept. [Wadler 2015]

Technical Writing

- **Exploring nullability in Kotlin**.
  - Highlighted the differential treatment of nullability in Kotlin and Java.
  - 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**
  - Explored challenges in implementing recursion within the interpreter.
  - Reviewed the mathematical theory of fixed-point combinators and thereby derived an implementation of recursion.
  - Demonstrated the use of fixed-point combinators concretely within the interpreter.

Skills

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