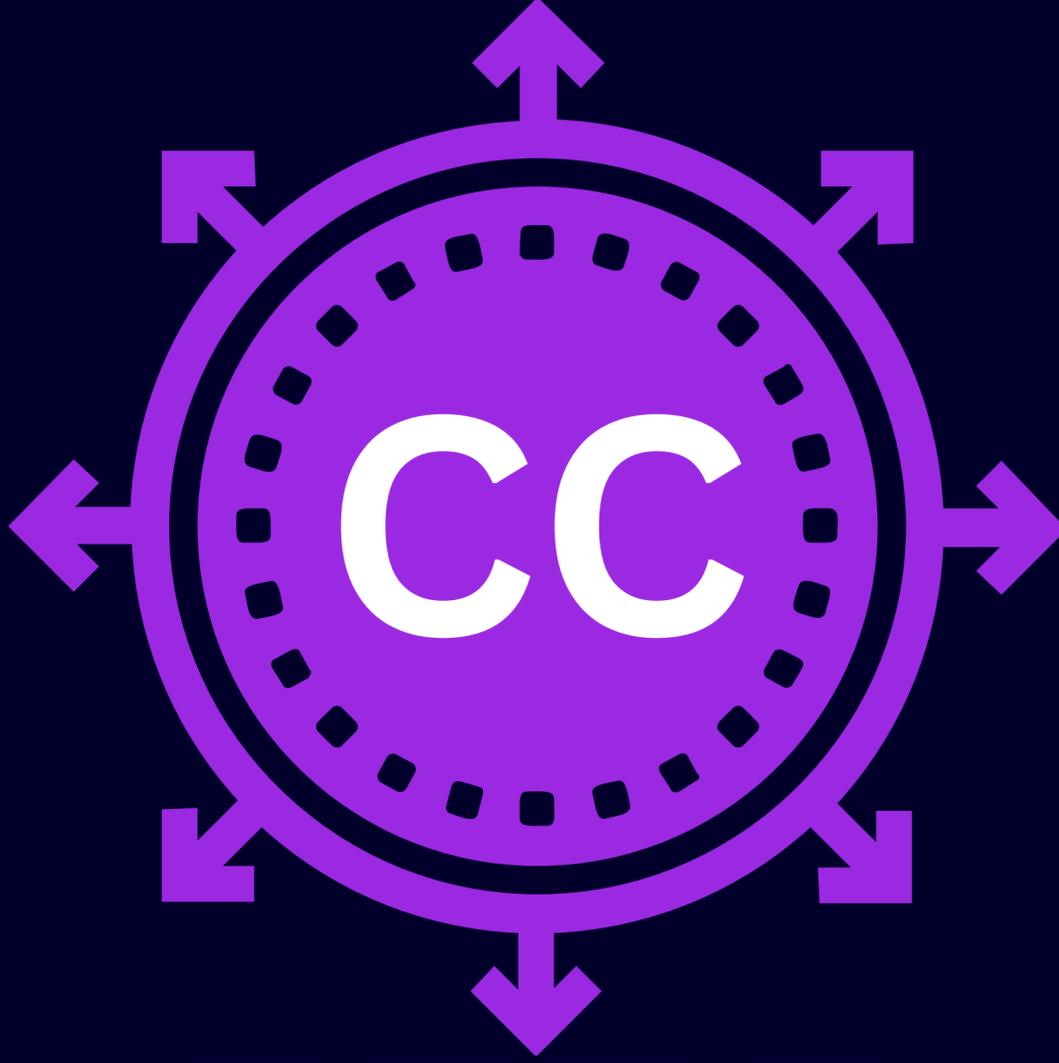




ಭಾರತೀಯ ಮಾಹಿತಿ ತಂತ್ರಜ್ಞಾನ ಸಂಸ್ಥೆ ರಾಯಚೂರು
भारतीय सूचना प्रौद्योगिकी संस्थान रायचूर
Indian Institute of Information Technology Raichur

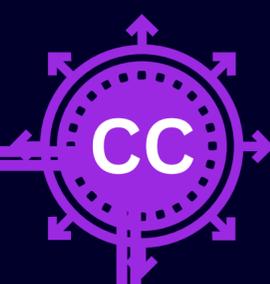


CODE CONFLUX

PROBLEM
STATEMENTS

12HR : 9AM TO 9PM

VENUE : IIITR ACADEMIC BLOCK



WEB DEVELOPMENT

1. Skill Exchange Platform for Students

Problem Statement: Students often have skills they want to share and others they want to learn, but finding the right match is challenging. A platform that connects students based on complementary skills can enable efficient peer learning.

Requirements:

- User registration with detailed skill profiles
- Search functionality to find suitable matches
- Chat feature for users to discuss and negotiate terms
- Review system to rate learning experiences
- Admin dashboard for platform monitoring

2. Online Compiler with Multi-Language Support

Problem Statement: Users need a convenient way to write, compile, and run code in multiple programming languages directly from a web interface without setting up environments locally.

Requirements:

- Code editor with syntax highlighting for multiple languages
- Server-side compilation and execution
- Real-time output display
- User authentication and profile management
- API integration for code execution services

3. Student Skill Trading Platform

Problem Statement: Many students have skills they want to trade (e.g., coding for photography lessons), but finding skill exchange opportunities is difficult. A platform facilitating skill trading can bridge this gap.

Requirements:

- Match users based on complementary skills
- Allow users to register, list offered/required skills, and set availability
- Integrated chat system
- Session scheduling and feedback system

4. Task Manager with Weather Integration & Portfolio Showcase

Problem Statement: Users need an all-in-one platform to manage tasks, track weather updates, and display personal achievements.

Requirements:

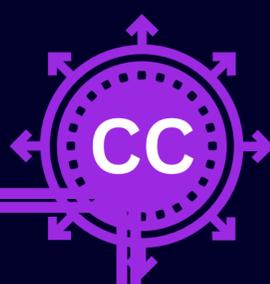
- To-do list for task management
- Real-time weather updates using OpenWeatherMap API
- Personal portfolio section for showcasing skills and projects
- Responsive design across all devices

5. Campus Navigation App

Problem Statement: Visitors and students often struggle to find locations on large campuses. An interactive navigation app can simplify movement within the campus.

Requirements:

- Interactive campus map
- Search functionality for buildings, departments, and landmarks
- Suggested routes with shortest path recommendations
- Integration with Leaflet.js or OpenStreetMap API



6. Student Portal for Centralized Information

Problem Statement: Universities use fragmented communication channels, making it difficult for students to access important updates, timetables, and announcements. A centralized student portal can solve this issue.

Requirements:

- Home page with student's name and quick links
- Announcements page for batch-specific updates
- Events page for university activities
- Timetable page for batch-specific schedules
- Profile page with student details
- Light/dark mode toggle
- Responsive UI using Bootstrap or Tailwind CSS

7. Accessible Resume Builder

Problem Statement:

Many online resume builders are cluttered, not mobile-friendly, and lack accessibility features for users with disabilities. This tool helps create clean, professional resumes with accessibility in mind.

Requirements:

- Responsive design for mobile and desktop
- Easy-to-fill form with live preview
- Accessibility features (keyboard navigation, screen reader compatibility)
- Option to download the resume as PDF
- Tech Stack: HTML, CSS

8. Minimalist Habit Tracker

Problem Statement:

Many habit tracker apps are overwhelming with too many features. This web app offers a simple, distraction-free way to track daily habits.

Requirements:

- Add, mark, and delete daily habits
- Clean, clutter-free interface
- Simple data persistence (local storage)
- Visual progress indicator (like streaks or checkmarks)
- Tech Stack: HTML, CSS, JavaScript

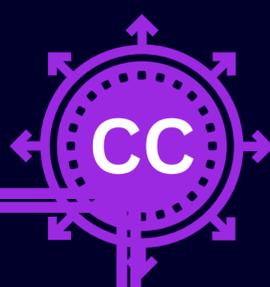
9. Tiny Web-Based Personal Finance Manager

Problem Statement:

People often find budgeting apps too complex. This tool provides a simple interface to track income, expenses, and savings without unnecessary features.

Requirements:

- Add income and expense entries with categories
- Dashboard showing total income, expenses, and balance
- Simple data visualization (like pie charts)
- Option to export data as CSV
- Tech Stack: HTML, CSS, JavaScript



10. Personal Productivity Dashboard (Static UI)

Problem Statement:

Many people want a centralized dashboard to visualize their tasks, goals, and schedule. This project creates a visually appealing static productivity dashboard.

Requirements:

- Sections for to-do lists, calendar view, and progress bars
- Aesthetic, clean layout with grid/flexbox
- Dark mode toggle
- Fully responsive design for desktop and mobile
- Tech Stack: HTML, CSS

11. Animated Infographic on Climate Change

Problem Statement:

Raising awareness about climate change requires engaging content. This project is an interactive infographic with animations to present key statistics effectively.

Requirements:

- Visually striking layout for data points
- Smooth CSS animations (charts, graphs, transitions)
- Scroll-based animations for dynamic storytelling
- Fully responsive for all screen sizes
- Tech Stack: HTML, CSS

12. Real-Time Weather Dashboard

Problem Statement:

Weather apps often display cluttered information. This project provides a clean, minimalist dashboard showing real-time weather data for any city.

Requirements:

- Input to search for cities
- Displays temperature, humidity, wind speed, and weather icons
- Dynamic background changes based on weather (e.g., sunny, rainy)
- Error handling for invalid city names
- Tech Stack: HTML, CSS, OpenWeatherMap API

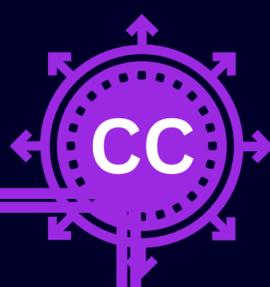
13. Crypto Price Tracker

Problem Statement:

Cryptocurrency enthusiasts often need quick price updates. This project provides real-time price tracking for popular cryptocurrencies.

Requirements:

- Displays real-time prices for Bitcoin, Ethereum, etc.
- Simple dashboard with percentage change indicators
- Refreshes data every minute automatically
- Dark mode for better viewing
- Tech Stack: HTML, CSS, CoinGecko API



14. Space Facts Explorer

Problem Statement:

Space enthusiasts love exploring facts about planets, galaxies, and missions. This project fetches real-time space data from NASA's public APIs.

Requirements:

- Displays NASA's Astronomy Picture of the Day (APOD)
- Section for interesting facts about planets and space missions
- Image gallery with information pop-ups
- Responsive design with smooth animations
- Tech Stack: HTML, CSS, NASA API

15. Public Holidays Finder

Problem Statement:

Finding public holidays for different countries can be tedious. This tool allows users to select a country and view its upcoming holidays.

Requirements:

- Dropdown menu to select a country
- Displays holidays with dates and brief descriptions
- Simple, clean UI for quick readability
- Error handling for API failures
- Tech Stack: HTML, CSS, Calendarific API

16. Movie Discovery App

Problem Statement:

Finding new movies to watch can be overwhelming with scattered recommendations. This app fetches trending movies with posters, ratings, and descriptions.

Requirements:

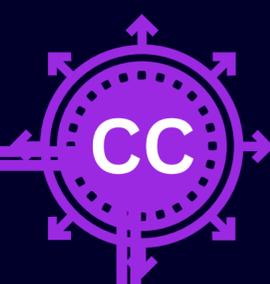
- Search bar to find movies by name
- Displays trending movies on the homepage
- Movie cards with poster images and ratings
- Clickable cards to view more details about each movie
- Tech Stack: HTML, CSS, TMDB (The Movie Database) API

17. Web for AI Tools

With the rapid growth of AI tools, finding the right one for a specific task can be overwhelming. A website that organizes AI tools based on their use cases will help users easily discover and select the best tools for their needs.

Requirements:

- Homepage Dashboard: Clean UI with search functionality for tasks (e.g., "Image Editing," "Text Summarization").
- Categorization of AI Tools:
 - a. Tools organized by categories like Text, Image, Video, Data Analysis, etc.
 - b. Subcategories for specific tasks.
- Recommendation Engine:
 - Suggest the best AI tools based on the user's input task.
- Tool Details Page:
 - Description, features, pricing, and usage tutorials for each tool.
- Integration & Links:
 - Direct links to access or integrate the tools.
- Responsive Design



18. Mathematical series and Algorithm visualiser and graph tool

Problem Statement:

Visualising how different algorithms work has been a difficult task. You are to create a website that makes this task much simpler, while you are free to add other features of your choice to improve the idea.

Requirements:

- User should be able to visualise the different series and algorithms that they want, while they are running the graph for the series, and the process for the algorithms.
- Changes in graph as per change in parameters of series and equations.
- Should be able to give equation for any curve drawn in the 2d graphical interface.

19. Home tools power usage tracker

Problem Statement:

Tracking how much power is being used by each device in our house is not easy, but optimising this device usage, wherever possible, through this approach, creates significant savings when bills are considered.

Requirements:

- User should be able to input all their house devices, their models, etc, and obtain analysis of how much power each device requires, as well as an overall estimate
- Users should also be able to add devices in different rooms (with this option being available from the get go), and track each room, or each set of devices' power usage separately as well.
- Other features and ideas that are in line with this idea or goal of optimising electricity usage in houses are encouraged as well.

APP DEVELOPMENT

1. Disaster Relief Coordination App

Problem Statement: During disasters, victims, NGOs, and volunteers struggle to communicate efficiently and allocate resources effectively. A platform that connects these groups can improve response efforts.

Requirements:

- Real-time location-based disaster mapping
- Resource request and allocation system
- Volunteer registration with task assignments
- Push notifications for updates and alerts
- Offline mode for critical information

2. Satellite Tracking App

Space enthusiasts and professionals may want to track satellites or space stations in real-time.

Requirements:

- A map displaying the locations of satellites, the ISS, and space stations in real-time.
- Information on each satellite, including its function, launch date, and country of origin.
- Alerts when a satellite or the ISS will pass over the user's location.
- The option to track specific satellites or space missions of interest.

3. Astronomy Observatory Scheduling App

Amateur astronomers or educators who want to visit observatories often lack an easy way to schedule visits or access equipment.

Requirements:

- Users can search for nearby observatories, their facilities, and available viewing equipment.
- Real-time schedule updates on upcoming public events or telescope time availability.
- Integration with telescopes or observatory cameras for remote observation capabilities.
- Calendar integration for easy booking of observation sessions or events.



AI/ML

1. Video Editing Tool using AI

Manually editing videos can be tedious. A tool that allows users to edit videos using simple text prompts will streamline the editing process, making it more intuitive and accessible.

Requirements:

- Input Video: Upload MP4, AVI, or other video formats.
- Text Prompts: Users provide simple text commands (e.g., "Add title at the start," "Insert subtitle at 30 seconds").
- Video Editing Actions: Process commands like cut, trim, add text, or transition based on the prompt.
- Preview: Show real-time preview of edits before final output.
- Export: Allow export in common formats (MP4, MOV).
- Text-to-Speech (Optional) and vice-versa: Generate voiceovers for textual prompts.
- User Interface: Simple input field for text commands, with feedback on each command(optional).

2. AI Cook

Cooking a dish can often be challenging without clear instructions and access to ingredients. An AI model that provides detailed cooking instructions, including a list of ingredients, images, and links to purchase them, will make cooking easier and more accessible.

Requirements:

1. Input Food Dish:
 - User inputs the name of the dish they want to cook or should give suggestions in the beginning
2. Ingredient List:
 - The model generates a detailed list of all ingredients required, including measurements.
 - Provide links to purchase each ingredient from popular online retailers(any third party)
 - Display images of the ingredients.
3. Step-by-Step Cooking Instructions:
 - Generate clear, detailed, step-by-step cooking instructions, formatted for easy understanding.
 - Include cooking times, temperatures, and any special techniques for each step.
4. Cooking Tips and Substitutes
5. Nutritional Information:
 - Display basic nutritional details of the dish (e.g., calories, proteins, fats).
6. Search Functionality:
 - Allow users to search for different recipes based on ingredients, cuisine type, or dish name.

3. AI Chatbot for Mental Health Assistance

Tracking expenses from receipts and bills can be cumbersome and error-prone. An AI tool that analyzes images of bills and receipts to extract the total money spent, along with detailed breakdowns, will help users efficiently track and manage their expenses.

Requirements:

1. Input:
 - Users upload images of receipts and bills (e.g., scanned, photos).
2. OCR-Based Text Extraction or other methods
3. Expense Breakdown:
 - Categorize the purchases (e.g., groceries, electronics, dining).
 - Display individual item prices, taxes, and discounts.
4. Total Money Spent:
 - Calculate and display the total amount spent for each receipt and for multiple receipts.
5. Visual Representation:
 - Provide a clear, interactive visual display of the breakdown of expenses (e.g., pie charts, bar graphs).
 - Include itemized lists with images of the receipt for context.
6. Data Export:
 - Allow users to export the expense details as CSV or PDF for further tracking.
7. Search & Filter:
 - Enable users to search receipts by vendor, date, category, or amount.
 - Filter expenses by category or time period



4. Autonomous Code Debugger using AI

Problem Statement: Debugging code can be time-consuming. AI can help automate the process by suggesting code corrections based on deep learning models.

Requirements:

- Input erroneous code for analysis
- Suggest corrections using a deep learning model
- Display the corrected code

5. Project Zamba: Computer Vision for Wildlife Research & Conservation

Problem Statement: Zamba is a tool built in Python that uses machine learning and computer vision to detect and classify animals in camera trap videos. It classifies species appearing in a video and can detect blank videos.

Requirements:

- Machine learning model to classify animal species
- Labeling and training of custom models for specific habitats
- Dataset of labeled wildlife video footage
- Python environment and machine learning libraries (e.g., TensorFlow, PyTorch)
- Website to showcase results (<https://zamba.drivendata.org/>)

6. Disease Spread Prediction

Problem Statement: Using environmental data, predict the number of dengue fever cases reported each week in San Juan, Puerto Rico, and Iquitos, Peru.

Requirements:

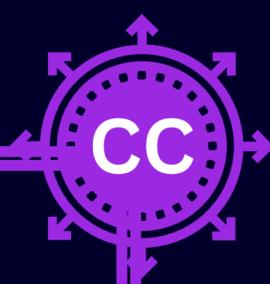
- Environmental data from multiple sources, such as CDC and NOAA
- Predictive machine learning model for disease case prediction
- Platform to showcase the model's predictions and results
- Link to data: [Dengue Prediction Challenge](<https://www.drivendata.org/competitions/44/dengai-predicting-disease-spread/page/80/>)
- Tools: Python libraries like Scikit-Learn or XGBoost

7. AI Doc Editor

Editing documents manually can be tedious and prone to errors. An AI-powered document editor that assists with content writing, formatting, and proofreading will enhance productivity, especially for long or complex text documents. This tool will automate common tasks like grammar checking, style improvements, and document structuring, making the editing process more efficient.

Requirements:

1. Input:
 - Users input or upload text documents (DOCX, TXT, PDF).
2. Text Prompt-Based Editing:
 - Users provide text-based commands for editing (e.g., "Correct grammar," "Make the tone more formal," "Change title to 'Chapter 1'") and AI will process these prompts and make corresponding edits in the document.
3. Grammar and Style Correction:
 - Correct spelling, punctuation, and sentence structure based on user input.
 - Adjust tone and style per instructions (e.g., formal, casual).
4. Formatting Changes:
 - Apply formatting changes as directed by text prompts (e.g., "Bold all headings," "Align text to the center").
5. Content Reorganization:
 - Move sections, paragraphs, or sentences based on commands (e.g., "Move the introduction to the end," "Merge these two paragraphs").
6. Export: Allow document export in multiple formats (DOCX, PDF, TXT).



BLOCKCHAIN

1. Blockchain-Based Certificate Verifier

Problem Statement:

Academic certificate fraud is a widespread issue, making it difficult for employers and institutions to verify the authenticity of credentials. Traditional methods are time-consuming, prone to human error, and vulnerable to forgery. A blockchain-based certificate verifier can ensure secure, tamper-proof, and easily verifiable academic records.

Requirements:

- User Roles: Admin (issuers), Students (certificate holders), and Verifiers (employers/institutions)
- Certificate Issuance: Universities issue digital certificates, hashed and stored on the blockchain
- Verification Portal: Public interface to verify certificates using transaction IDs or QR codes
- Blockchain Integration: Use Ethereum or Polygon with smart contracts for certificate storage
- Data Privacy: Sensitive data stored off-chain with hashes on-chain to ensure privacy
- Audit Trail: Immutable record of certificate issuance and modifications

2. Blockchain-Based Voting System

Problem Statement: Election integrity is often questioned due to tampering and lack of transparency. A blockchain-based voting system ensures secure, immutable, and transparent elections.

Requirements:

- Digital identity authentication
- Blockchain ledger for immutable vote storage
- Smart contracts for secure voting logic
- Real-time vote count display
- Mobile/web-friendly user interface

GAME DEV

1. Memory Maze

Memory games have always been interesting, but never has memory been relied upon to clear a maze, which is the epitome of path unpredictability. You are required to create a game that combines memory evaluation with maze clearing aspects, with the following features.

Requirements:

- At the beginning, a random maze is generated and a user should get a 10 second top view of the game.
- The maze should then transform into a front-view (first-person view for the player).
- The player is to navigate this maze and clear it.
- Different times of clearing the maze gives different scores, based on which high scores are also listed.
- There can also be different levels of the difficulty, and separate high scores for each of them, with differing amounts of time which the player is allowed to see and memorise the top view of the maze.