IITISoC' 23

Problem Statements

Index

1 AI/ML 2

2 Analytics 7

3 App Development 9

4 Astronomy 14

5 Intelligent Vehicles and Robotics 15

6 Blockchain 18

7 Design and Modelling 20

8 Electronics 21

9 Extensions 23

10 Finance 25

11 Game Development 27

12 Material Design and Process Optimization 29

13 Quantum Computing 30

14 Web development 31

15 Cybersecurity 38

16 On-site Concrete Technology 41

AI/ML

1. Lip Reading AI Beginner

Description: Build a model designed to provide subtitles for silent videos featuring talking individuals

Specification:

- Build an innovative model designed to provide subtitles for silent videos featuring talking individuals. With a focus on assisting people with hearing impairments, this technology utilizes computer vision and natural language processing techniques to generate accurate and synchronized subtitles, ensuring an inclusive viewing experience. Bonus Points for Creating an App or Website to showcase the finished product.

2. Literature Review Generator Beginner

Description: Build a model to review scientific papers on a daily basis and generate concise summaries for a specific domain.

Specification:

- Build a system to review scientific papers on a daily basis and generate concise summaries in a specific domain. the model should analyze the papers' content, including abstracts, introductions, and key sections. It identifies important concepts, methodologies, findings, and conclusions within each paper. Based on this analysis, the model generates concise summaries that capture the essence of the research, highlighting key contributions and insights. Bonus Points for Creating an App or Website to showcase the finished product.

3. Musical Tone Generator Beginner

Description: Create a Musical Tone Generator capable of analyzing existing music data and generating new and unique musical tones in the form of musical notes

Specification:

- Create a Musical Tone Generator capable of analyzing existing music data and generating new and unique musical tones in the form of musical notes. You can also create a user-friendly interface that allows musicians and composers to interact with the system. The interface should enable users to input preferences, explore generated musical tones, modify parameters, and integrate the generated tones into their creative workflows. Bonus Points for Creating an App or Website to showcase the finished product.

4. Noice Cancellation Beginner

Description: Develop a model to eliminate background noise from recordings.

Specification:

- Develop an effective Deep Learning-Based Noise Reduction Model for call recordings, aimed at addressing the challenge of background noise interference and delivering clear, noise-free audio output. The objective is to enhance the listening experience, ensure optimal communication clarity, and improve the overall quality of call recordings in various environments and communication scenarios. Bonus Points for Creating an App or Website to showcase the finished product.

5. Song Recommender System Beginner

Description: Develop a recommendation system to recommend songs based on user preference and history.

Specification:

- Develop a model to capture user preferences, considering factors such as favorite genres, artists, albums, moods, and specific song attributes like tempo, instrumentation, or lyrical themes of their preferred songs. Efficient and user-friendly interfaces or algorithms need to be designed to gather this information from users. developing algorithms that can leverage user preferences, historical data, collaborative filtering techniques, and content-based analysis to generate personalized recommendations. Bonus Points for Creating an App or Website to showcase the finished product.

6. Air Canvas Beginner

Description: Develop an Art Canvas on which user can draw using the camera input.

Specification:

- The problem is to develop a real-time fingertip detection and virtual canvas drawing system using the device's camera. The challenge is to accurately detect and track the fingertip, enabling seamless drawing on a virtual canvas, while ensuring responsive and smooth rendering for an intuitive and enjoyable user experience. Develop a Website or App for this tool.

7. Movie Suggestor Beginner

Description: Recommend Movies based on their plots.

Specification:

- Develop a movie recommendation system that analyzes movie plots and identifies story similarities to suggest similar movies. The challenge is to extract meaningful features from movie plots, apply effective similarity measures, and create a scalable and accurate recommendation system to enhance the movie discovery experience for users. Bonus Points for Creating an App or Website to showcase the finished product.

8. Speed Cam Beginner

Description: Develop a Model to measure speed of a vehicle from a video of it moving on the road.

Specification:

- Develop an object detection system capable of accurately predicting the speed of running cars based on visual data. The challenge is to design and train a model that can detect and track cars in real-time, extract relevant motion features, and accurately estimate their speed, enabling applications such as traffic management and surveillance. Bonus Points for Creating an App or Website to showcase the finished product.

9. Image Gallery Organizer Intermediate

Description: Make a MultiLabel Model to classify and oraganize images.

Specification:

- Design an image grouping system that categorizes a set of images based on user-defined criteria such as person, color, category, or nature. Develop a machine learning model capable of analyzing image features and organizing them into distinct groups, facilitating efficient image organization and retrieval for users. Bonus Points for Creating an App or Website to showcase the finished product.

10. Image Story Generator Intermediate

Description: Make a model to generate a story from a given input image.

Specification:

- Develop an image summarization model coupled with prompt engineering techniques that can generate a coherent and engaging story based on a single input image. The challenge is to extract relevant visual information, leverage text generation methods, and incorporate contextual prompts to create a compelling narrative that aligns with the content and context of the image. Bonus Points for Creating an App or Website to showcase the finished product.

11. Image Deblurring Intermediate

Description: Make a Model to remove blur from images. which might be blurred due to jerky motion while the image was captured.

Specification:

- Design and implement a convolutional neural network (CNN) for image deblurring, utilizing specialized filters and iterative techniques. Train the model on a diverse dataset to learn and restore image details, enhancing visual clarity and sharpness. Bonus Points for Creating an App or Website to showcase the finished product.

12. Object detection using Lidar Data Intermediate

Description: Make a Model to detect objects using LIDAR data.

Specification:

- Develop a deep learning framework that utilizes LiDAR data for 3D object detection in autonomous vehicles. The framework should accurately identify and classify objects, including cars, pedestrians, and obstacles. Train the model on diverse LiDAR datasets to enable reliable and real-time perception of the surrounding environment, ensuring safe and efficient autonomous navigation

13. Mario Playing AI Intermediate

Description: Train an Agent to play Mario.

Specification:

- Utilize reinforcement learning (RL) to develop an agent capable of autonomously playing Super Mario by learning from interactions with the game environment. The objective is to train the agent to navigate through levels, avoid obstacles, collect power-ups, and successfully complete the game, achieving proficient gameplay performance.

14. Speech Emotion Detection Intermediate

Description: Detect Emotion from speech by the tone of the voice.

Specification:

- Develop a robust system for emotion detection from speech, using machine learning techniques. The system should accurately classify emotions such as happiness, sadness, anger, and more, based on the acoustic features and patterns present in the speech signal. Train the model on diverse emotional speech datasets to ensure generalizability and real-world applicability. Bonus Points for Creating an App or Website to showcase the finished product.

15. False Information Detector Advanced

Description: Develop a model to take in the thumbnail, title and description of a YouTube video and classify it as misinformation or legitimate information.

Specification:

- Train a multimodal model to detect the presence of false information in YouTube videos by analyzing their thumbnail, title, and potentially their subtitles or descriptions. The challenge is to develop an effective model that can extract meaningful features from these modalities and accurately classify videos as containing or not containing false information, contributing to the fight against misinformation on online platforms. Bonus Points for Creating an App or Website to showcase the finished product.

16. Type in your handwriting! Advanced

Description: Create a model to learn a handwriting, and allow you to generate more by simply typing

Specification:

- Create a machine learning model that learns and mimics personal handwriting style. The model should be trained on a dataset of handwritten samples of a person and capable of converting standard fonts into any given person's personalized handwriting. The objective is to generate handwritten text that closely resembles unique writing style for various applications. Bonus Points for Creating an App or Website to showcase the finished product.

17. Guide Dog AI Advanced

Description: Create a Model to inspect the surroundings using camera and navigate to the provided location using GPS.

Specification:

- Blind People generally use guide dogs to move around. Create a system that can scan the surroundings and take constant video input and GPS location. It should then use path-finding and object detection to provide instructions on how to navigate vocally. It should also include a system to instruct the directions and/or describe obstacles in the way of a location. Bonus Points for Creating an App to showcase the finished product.

Analytics

1. Music Snowflake Intermediate

Description: Create a Music Visualizer Platform that takes in any kind of music as an input and subsequently visualizes it as it plays.

Specification:

- The imagery/visuals should be generated and rendered in real time and in a way such that it is synchronized with the music as it is played.
- Also add a model which initially detects the Genre of Music taken as an input. Tags:Python,Data Visualization,ML

2. Machine-Algo Visualizer Intermediate

Description: Create a platform that visualizes how a machine learning algorithm runs on a particular dataset and fits on the data.

Specification:

- It should visualize the decision boundary of various machine learning algorithms (SVM, decision trees,etc)
- Users should be able to change various parameters related to the algorithm and observe the change on the data.

3. BigQuery runner and Graphing Dashboard Intermediate

Description: Create a Web portal for running queries and creating graphs

Specification:

- User must be able to upload and download dataset in form of CSV
- The user must be able to pass SQL queries on the dataset
- The web portal must create automated graphs depending on the type of attributes in data such as nominal, ordinal or cardinal.

4. Supply Chain forecasting Intermediate

Description: Create a Dashboard for time series forecasting

Specification:

- Use Tableau or PowerBI for the graphs and other entities.
- The app should use supply chain analytics to propose optimisation strategies.

- The app should also forecast future trends in the supply chain.

5. Inventory Optimization Intermediate

Description: Create a web portal where the optimum amount of commodities will be displayed based on their demand.

Specification:

- The web portal should create automated graphs which should display the trends in demands of the commodities in past and the expected trend in the future. Tags: Data Visualisation, Python, ML.

6. Energy Consumption Forecasting for Smart Homes Advanced

Description: Create a platform backed by a model that predicts energy consumption patterns for smart homes based on historical data, weather conditions for optimizing their energy usage.

Specification:

- The model should get automatically updated as new data is collected and provided to it.
- Users must be able to visualize the insights from data through the interface.

App Development

1. LoFo App Beginner

Description: Create a lost and found application. Lost and found website fails its purpose as does not provide notifications to the users. An application will remove this loophole. also include features like buying and selling. Application should be cross platform. (Hence flutter, react native are prefered)

Specification:

- A user should be able to Add photographs of any found item, which other users can claim.
- A user can add their lost item in the lost section which other users can view.
- A user can post items they want to sell, which other user would be able to buy.
- Posting an item in any section, should ask the user for some basic required informations about the item, to confirm genuity of it.

2. Student Life App Beginner

Description: Design and develop an intuitive Android application named "Student Life" for the students of IIT Indore, providing them with a centralized platform to receive timely notifications about all events and programs held within the institute. The "Student Life" app aims to enhance student engagement, foster a sense of community, and streamline event communication by providing a user-friendly interface, timely notifications, and interactive features for students at IIT Indore.

Specification:

- Students often struggle to stay updated on various events and programs happening within the institute. There is a need for an app that consolidatesall event-related information and notifications in one place for easy access and awareness.
- The app should provide a reliable and efficient medium for event organizers to notify and engage with the entire student community.
- The app should offer a user-friendly interface that allows the Categorization of events into various categories (e.g., academic, cultural, sports, social), enabling students to filter and explore events based on their interests.
- The app should include the feature of Instant notifications about upcoming events, programs, workshops, seminars, and competitions held within the institute.
- The app should include comprehensive information about each event, including the date, time, venue, description, and any registration requirements.

3. Weather App Beginner

Description: The objective of this project is to create a simple weather app that provides users with current weather information for a specific location.

Specification:

- Attractive User Interface: The App should have a good user interface.
- Wheather Details Provider: The app should be able to access the location of the user and provide them with he temperature, weather conditions, and additional details such as humidity, wind speed, and forecast for the upcoming days.
- Global Search: It should also provide a search feature that enables users to access whether details around the globe.
- Notification Provider: The app should notify users of today's weather every morning.

4. E-Vehicle and Bus Tracking App Beginner

Description: This project would serve as an important step to help in Vehicle tracking. As a beginner, the task is to create a mobile application that allows users to track the location and availability of E-Vehicles and buses within the campus of IIT Indore. The app should provide a user-friendly interface and deliver real-time information to enhance transportation efficiency and convenience for the students, faculty, and staff of the institute.

Specification:

- Vehicle Availability:
  - Users should be able to quickly check the availability of E-Vehicles and buses
  - The app should display the number of available E-Vehicles and buses, in real-time. indicating whether they are currently in service or unavailable.
- E-Vehicle Tracking:
  - Users should be able to view the real-time location and availability status of E-Vehicles on the campus.
  - The app should display the E-Vehicles on a map, allowing users to see their current positions.
- Simple User Interface: The app should have a user-friendly interface with clear navigation and minimalistic design.

5. Music Streaming App Intermediate

Description: Create a music streaming app that provides users with access to an extensive collection of songs, personalized playlists based on their music preferences, and the ability to share their favorite music with friends and followers. Enhance the music listening experience by offering a seamless and intuitive interface, curated recommendations, and social sharing features to connect music enthusiasts worldwide.

Specification:

- Vast music library: Offer a comprehensive collection of songs spanning various genres and languages.
- Personalized playlists: Generate custom playlists for users based on their music preferences, listening history, and favorite artists or genres.
- Seamless user interface: Design an intuitive and user-friendly interface for easy navigation and smooth music playback.
- Curated recommendations: Provide personalized song and artist recommendations to help users discover new music based on their interests and listening habits.
- Social sharing features: Enable users to share their favorite songs, playlists, and music discoveries with friends and followers through social media integration.
- Offline playback: Allow users to download songs and playlists for offline listening, ensuring uninterrupted music enjoyment even without an internet connection.
- User profiles: Provide users with the ability to create profiles, customize their music preferences, and follow other music enthusiasts for a more social experience.
- Search functionality: Implement a robust search feature that enables users to find specific songs, albums, or artists quickly and accurately.
- High-quality audio streaming: Ensure the app supports high-quality audio streaming to deliver an immersive and enjoyable music listening experience.
- Cross-platform compatibility: Develop the app to be compatible with both iOS and Android devices to reach a wider user base.

6. Food Delivery App Intermediate

Description: This food delivery app is designed specifically for students, offering a convenient and affordable way to satisfy their cravings. With a curated selection of local restaurants popular among students, the app allows users to browse menus, customize their orders, and easily place them with just a few taps. Students can track their orders in real-time, receiving updates on the status and estimated delivery time. We also provide a range of payment options, including cash on delivery and mobile wallets, making it hassle-free for students to complete their transactions. Whether it's a late-night study session or a weekend hangout, our food delivery app is here to ensure that students have access to delicious meals whenever they need them.

Specification:

- User Registration and Authentication:
  - User registration and login functionality using email or social media accounts.
  - Simple and streamlined authentication process with password recovery option.
- Restaurant Listing and Search:
  - Display a curated list of local restaurants popular among students.
  - Implement a search feature to find restaurants based on cuisine type, location, or ratings.

  - Allow filtering options to cater to specific dietary preferences (e.g., vegetarian, vegan, halal).

- Menu and Ordering:
  - Present menus with clear item descriptions, prices, and any available student discounts or deals.•
  - Enable users to add items to a cart and customize orders (e.g., toppings, spice levels)•.
  - Support a straightforward ordering process with a clear checkout interfac.
- Real-Time Order Tracking:
  - Provide basic order tracking with status updates (e.g., order confirmed, preparing, out for delivery).
  - Send notifications to users about their order status changes.
  - Display an estimated delivery time based on the restaurant's average delivery times.
- Payments and Checkout:
  - Offer popular payment options such as cash on delivery or online payment through mobile wallets.
  - Implement a secure payment gateway with encryption for transactional data.
- Ratings and Reviews:
  - Allow users to rate and provide reviews for restaurants and their dishes.
  - Display average ratings and reviews for each restaurant to assist students in making informed choices.
- Delivery Information:
  - Collect basic delivery information, including the student's dormitory or residential address.
  - Store user preferences for future orders, such as delivery instructions or preferred delivery times.

7. Dietary App Intermediate

Description: Individuals seeking to improve their nutrition and adopt healthy eating habits face challenges in finding a user-friendly and personalized solution. Existing nutrition apps lack the simplicity and customization needed to provide personalized dietary recommendations, diverse meal ideas, and seamless nutritional tracking. To address these challenges, our proposed nutrition and meal planning app will deliver personalized dietary recommendations, a wide variety of meal ideas, and streamlined nutritional tracking in a single, user-friendly interface.

Specification:

- East to understand UI: Provide a user-friendly, easy-to-use interface for the app.
- Personalized Recommendations: This app should provide users with personalized dietary recommendations based on their unique factors such as age, gender, weight, activity level, nutritional restrictions, and health goals.

- Inclusive to all Diet types: The app should offer various meal ideas that cater to different dietary preferences (e.g., vegan, vegetarian, gluten-free) and include multiple cuisines.
- Nutritional Tracking: The app should have a nutrition tracking system that enables users to log their food intake and track macronutrients. The app will provide a primary database of commonly consumed food items and their nutritional information to facilitate efficient tracking.
- Deit Coach: The app should have a feature that allows users to input a time period and a desired weight along with their current weight and then calculates the daily calorie intake required to reach the desired weight within the specified timeline. It should also remind them to input their diet daily and tell them if the diet is within the specified limit or not.

8. Language learning app Advanced

Description: The challenge is to develop an advanced language learning app that offers users an immersive and interactive language learning experience. Existing language learning apps often need more personalized feedback on speaking and pronunciation skills, making it challenging for users to improve their oral proficiency. Additionally, many language learning apps need more engaging elements to keep users motivated and actively engaged in the learning process. Therefore, there is a need for a language learning app that incorporates speech recognition technology for spoken practice and provides gamification features to enhance user engagement and motivation.

Specification:

- Speech Recognition: The app should incorporate speech recognition technology to enable users to practice speaking and improve pronunciation. It should utilize APIs or libraries that provide speech-to-text capabilities to transcribe and analyze users' spoken language.
- Personalized Feedback: The app should provide personalized feedback and suggestions based on the accuracy and fluency of users' pronunciation, allowing them to identify areas for improvement and track their progress over time.
- Interesting UX: The app should include gamification elements such as quizzes, challenges, and rewards to enhance user engagement and motivation. Users can participate in interactive language exercises, vocabulary quizzes, and pronunciation drills.
- Reward System: Progress tracking and achievement systems should be implemented to provide a sense of accomplishment and encourage users to continue their language learning journey.
- Detailed UI: The user interface should be designed intuitively, providing learners with a seamless and immersive experience. It should include lesson modules, vocabulary lists, speaking practice exercises, and progress tracking.
- Aditional Features: The app may offer additional features like language-specific cultural insights, grammar tips, and community forums where users can interact with fellow learners or native speakers.

Astronomy

1. Exohunt Intermediate

Description Design a webpage that facilitates the retrieval of exoplanet details taking the star's ID(from any catalogue) as input. The backend program should be capable of searching for a suitable target pixel file from TESS and Kepler projects(from the MAST Archive) based on the object ID, plotting its lightcurve, and extracting important parameters such as size and masses of the star and the planet and planet's orbital period. These parameters should be included in the output.

Specification:

- The goal of this task is to create a user-friendly webpage that enables users to explore exoplanet details using data from the TESS and Kepler projects.
- The website should have a simple and intuitive interface where users can input the name of the desired object(present in the catalogues).
- The backend program should utilize the provided object ID to search for a suitable target pixel file from the MAST Archive which hosts raw and processed data from TESS and Kepler projects. From this file, the program should generate a lightcurve, representing the flux of the target over time, and analyze for parameters such as, size and mass of the star and planet and orbital period of the Exoplanet present.
- The website should display the lightcurve graphically, allowing users to visualize the data. Additionally, the backend program should also return the mentioned parameters onto the webpage.

2. Exploring Elliptical Galaxies Advanced

Description: Using observational data, study elliptical galaxies, their classification, and radial light distribution.

Specification:

- The project uses observational data to research and learn about elliptical galaxies, their classification, and the radial distribution of their surface brightness. You will be given the required data. You must measure galaxy sizes along the major axes for classification and analyze the radial light distribution based on de Vaucouleurs' law.

Intelligent Vehicles and Robotics

1. Traffic Light Negotiation and Perception-Based Detection (Example) Beginner

Description: Use MATLAB® and Simulink® to load a pre-built Unreal scene, detect and identify the state of the traffic light nearest to the ego-vehicle, design a Stateflow® model to control traffic lights in the scene, and control the reaction of the ego-vehicle in accordance to the traffic lights and surrounding vehicles.

2. Make a Pet[Quadruped Robot](Example Library) Beginner

Description: Import a CAD model of the Quadruped robot in Gazebo and use relevant MathWorks packages and open-source libraries to control its movement. It must be able to perform certain pre-determined tasks involving human-robot interaction. You may get a chance to use your code on a similar quadruped robot for the Robotics Club(post-IITISOC).

Specification:

- Quadruped Robots, Control, Path planning, Human-Robot Interaction

3. Lane detection using limited computation power Intermediate

Description: Autonomous Vehicles require advanced lane detection algorithms to perceive lanes on the road(Street as well as rough road lanes). Develop a robust lane detection pipeline that consumes meager computational resources (No GPU allowed, limited CPU and RAM usage) and could be deployed on NVIDIA Jetson Nano Board or even a Raspberry Pi board. If this project is successful, you will be involved in actually implementing your software on a real-world vehicle (in the autonomous vehicle team of the IVDC club), especially for IGVC International Competition

Specification:

- Computer Vision, DL, Pytorch/Tensorflow

4. Path Planning for Autonomous Race Cars (GitHub)
Intermediate

Description: Finding an optimal racing line can be considered an optimization problem. Work with MATLAB® and the Optimization Toolbox™ to develop the algorithm.

Specification:

- Autonomous Vehicles, Optimization Techniques/ RL, Modeling and Simulation , MATLAB

5. Design of Humanoid Robot(Example)
Intermediate

Description: Design a humanoid robot based on your simulation results. Outline the various requirements for advanced functions like walking, identifying objects/markers and reacting accordingly. You may use your results and code to design Robotics Club's Humanoid Project(post-IITISOC).

Specification:

- Humanoid Robot, Control, Path planning, Human-Robot Interaction, Bipedal Motion, Reinforcement Learning

6. Underwater Drone Hide and Seek(Example)
Intermediate

Description: Design algorithms for underwater drones to compete in a hostile situation, where a first drone is trying to stay stealth and a second drone is actively searching for the first drone. Develop an algorithm to navigate a stealth underwater drone without being detected. Develop another algorithm to navigate a second underwater drone to search for the stealth drone.

Specification:

- Underwater Drones, Control, Optimisation, Game Theory/RL, Military

7. Intelligent Path Planning for Quadrotors(Github)
Intermediate

Description: When palnning trajectory, it is not only important to plan a safe route, but also which saves resources. Work with MATLAB® and the Optimization Toolbox™ to develop algorithms for best path planning which saves time, resources and satisfies safety constraints.

Specification:

- Drones, Robotics, Autonomous Vehicles, Electrification, Modeling and Simulation, Optimization, UAV, MATLAB

8. Predictive Electric Vehicle Cooling(GitHub) Advanced

Description: Work with Simscape™ Fluids™ to create a plant and predictive controller for EV cooling system with dynamic loads using MATLAB and Simulink. The model should be detailed enough to capture important dynamics. Dynamic loads include outside environmental conditions, fast charging, and rapid acceleration/deceleration. Demonstrate that the predictive control system can keep battery temperature in the desired range. Demonstrate whether the control can allow for a greater performance envelope for motor loads and fast charging. Compute the change in energy requirement from operating the cooling system predictively vs. reactively. Extend the work to predict battery range and life expectancy improvement with the predictive controller.

Specification:

- Sustainability and Renewable Energy, Autonomous Vehicles, Automotive, Control, Electrification, Modeling and Simulation, Optimization

9. Autonomous Navigation for Vehicles in Rough Terrain[Space] (GitHub) Advanced

Description: Demo a robot/vehicle (AMR, front loader, excavator, curiosity mars rover) working in a cluttered field (off-road) moving from point A to point B. The field should have minor bumps and ditches

Specification:

- Autonomous Vehicles, Computer Vision, Robotics, Image Processing, Mobile Robots, SLAM, UGV, Optimization , MATLAB

10. Warehouse Robotics Simulation(Example) Advanced

Description: This project focuses on the design and simulation of warehouse robotic operations. Using the functionality available in perception, navigation, fusion, and applications products listed like, Computer Vision Toolbox™, Navigation Toolbox™, Sensor Fusion and Tracking Toolbox™, Robotics System Toolbox™, etc. to simulate the warehouse operations and ascertains their efficiency and suitability for the task at hand.

Specification:

- Modelling and Simulation, Multi-Robot Systems, Control, Optimization, Computer Vision, Path-Planning/Robot Arm Trajectory Planning(based on which robot you choose in the proposal)

Blockchain

1. To-do List Application Beginner

Description: Create a to-do list app that uses blockchain technology to store data.

2. Blockchain Messaging System Beginner

Description: A blockchain-based messaging system uses blockchain technology to store and transmit messages.

3. Decentralized Polling System Beginner

Description: Create a polling system that allows users to create anonymous polls and vote on various polls.

4. Collectible Creatures Intermediate

Description: Create a blockchain-based game where players can collect and trade unique virtual creatures as NFTs. Each creature could have its own attributes, abilities, and rarity, making them valuable and sought after. Players can battle their creatures against others, level them up, and even breed new and rare creatures.

5. Decentralized Cryptocurrency Exchange Intermediate

Description: This would be a non-custodial cryptocurrency exchange that means it won't be live on a server somewhere. It will allow you to own and control the private keys to your cryptocurrency. There are so many cryptocurrency users out there and they all want to trade their cryptocurrency at some point in time and a decentralized cryptocurrency exchange would allow them to do this. An example of such an application is Idex. Now how to build this decentralized cryptocurrency exchange? To build this, you would need to create some smart contracts that handle the trading. Your application should be able to trade any type of ethereum assets, ethereum cryptocurrency itself, and any ethereum based tokens. Once you have the smart contracts in place to do this you would need some sort of client-side interface so that people can interact with your application and use it. For making the interface, you can use any backend framework like React.js, Node.js, Angular, Django, etc. Now in order to convert your web application into a blockchain application, you'll also need some sort of library. A library that turns your web application into a blockchain website and that's where web3.js come. Web3.js allows you to talk to smart contracts inside a regular website.

6. Decentralized Identity Management Intermediate

Description: Create a blockchain-based solution that enables individuals to have full control over their digital identities, allowing them to manage and share their personal information securely and selectively.

7. Blockchain-powered Freelance Marketplace Intermediate

Description: Create a decentralized platform that connects freelancers directly with clients, eliminating intermediaries, ensuring secure and transparent transactions, and facilitating fair dispute resolution.

8. Decentralized HealthCare Data Record Advanced

Description: Build a privacy-preserving blockchain solution for managing and sharing medical records. That will help patients to share their medical records across various hospitals and health care centers.

9. Carbon Credits and Emissions Tracking Advanced

Description: Create a blockchain solution for tracking carbon credits and emissions data, providing a transparent and immutable record of carbon offsets and facilitating the trading of carbon credits.

10. Blockchain Treasure Hunt Advanced

Description: Design a game that involves a virtual treasure hunt where players search for hidden treasures represented as NFTs. Each treasure can have different rarity levels and unique characteristics. Players can solve puzzles, explore virtual worlds, and collaborate with others to find and collect these treasures.

Design and Modelling

1. Rocket Reentry Steering Intermediate

Description Design a Rocket re-entry steering(flaps) mechanism that enables controlled and precise steering during the reentry phase, ensuring a safe and accurate landing. The mechanism should have strong structural integrity, optimised aerodynamic performance, and be validated through simulations under various reentry conditions.

Specification:

- Given: Maximum height attained by the rocket - 3000 m Maximum horizontal velocity at highest point - 5 m/s Mass of the rocket - 20 kg Deliverables: Detailed CAD model of the rocket reentry steering(flaps). Animation showing the working of the mechanism. CFD simulations, FEA simulations Modal Analysis. Note: This project requires CAD design skills, knowledge of aerodynamics, control systems, and simulation techniques to achieve controlled reentry and landing of the rocket.

2. 3D model and CFD analysis of Hostel Intermediate

Description: Reconstruct our hostels and use computational fluid dynamics (CFD) study to enhance the natural ventilation of the hostels. You can also use CFD analysis to determine which unit number (room number) has the best natural airflow and to suggest improvements. Likewise, don't forget the badminton court. Some of the elements that will be taken into account in the CFD analysis include the following:

- The size and shape of the hostels.
- The location of the hostels.
- The surrounding environment.
- The number of occupants.
- The activities that take place in the hostels.

Specification:

- The cost of the improvements.
- The effectiveness of the improvements.
- The impact of the improvements on the residents.

Electronics

1. ALU design Beginner

Description: Design a simple Arithmetic Logic Unit using Hardware Description Language.

Specification:

- The heart and core of any processor is its arithmetic logic unit. The ALU does almost all the job that are computations to be done on input data. You have to use hardware description language (HDL) to construct an ALU for a processor that can do following functions on the input data. Here, You can consider a ALU which takes two numbers(A and B) with input bits as 8 and 6 bit control signals of 6 bits and gives a 8 bit output. control signals - first bit(LSB) - zeroes all the bits of A second bit - negates A third bit - zeroes all the bits of B fourth bit - negates B fifth bit - if 0 - output = A+B if 1 - output = AB sixth bit(MSB) - negates output All the control signal bits are executed sequentially. functions outputs that are required to be done with ALU are - 1) 0 2) 1 3) -1 4) A 5) A 6) B 7) B 8) A+B 9) AB 10) A-B 11) B-A 12) A-1 13) B-1 14) A|B 15) A+1 16) B+1 17) -A 18) -B 19) (A+B) 20) (AB)
  You have to find control signals such that they perform this task. You have to write the instructions in Verilog language only.

2. Design and write a Device Driver Intermediate

Description: Design a simple device driver(include as much as features as you can) of your choice for any microcontroller or microprocessor. WITHOUT USING ANY PRE-EXISTING LIBRARIES OR DRIVERS

Specification:

- PS: make libraries for a microcontroller to write in a SD card, or any such complex functionalities in embedded C(simple microcontroller programming is in C). The question is open ended. Writing functions and libraries for writing data in SD card is an example you are free to choose any similar ones like BLUETOOTH or WIFI libraries of arduino microcontrollers or any such manufacturer. These modules can be found in simulation tools, you are requested to find the tools which are convenient for you and design your custom libraries and hence make your own device driver. (Eg of simulation tools: TinkerCad,etc). You can use IDE tools like STM32CubeIDE, Arduino IDE. Do not use any pre-existing libraries or drivers
  NOTE: The below is a very simple example. Submitting this is higly discouraged. EXAMPLE: For a motor controller shield by Arduino, there exists a function called setspeed(values from 0 to 255). The function goes and fetches the value from the table(values inputted vs voltage supplied to motor) and changes the voltage using PWM(by Duty Cycle modification. Vout= Dvsource, you can understand this if you learn about buck or boost converters online). It

also chooses an optimal frequency for the PWM of the motor depending upon the inducatance values(that's why table is required it won't be a directly proportional always like formula suggests). Then it provides the required voltage. Also it maintains sufficient torque(torque is proportional to current, table required here too) to maintain a specific speed. You are requested to design these kinds of libraries, drivers yourselves, without using any pre-existing libraries or driver.

NOTE:plagiarims check will be done for the libraries.

3. Microprocessor Design Advanced

Description: Design the microprocessor of a computer using Hardware Description Language

Specification:

- Hardware description languages are one of the most essential tools in designing digital components like Registers, ALU, CU of a computer. Design a microprocessor using Verilog language. The students have the freedom to select the type of architecture for their microprocessor. The bit-size of memory, ALU and other units of the CPU are not fixed. Remember, the design should be in Verilog language only.

Extensions

1. Note Making Chrome Extension Intermediate

Description: The extension allows users to capture important information, jot down thoughts, or save snippets of text for future reference. It offers a user-friendly interface, seamless integration with the Chrome browser, and synchronization across multiple devices.

Specification:

- Users can create, edit, and delete notes directly from the extension popup. Notes can be organized into categories or tags for easy management.
- The extension provides a text editor with essential formatting options, such as font styles, text color, and alignment, making it easy to customize the appearance of notes.
- Users can quickly find specific notes by searching for keywords or applying filters based on tags or categories.
- For reference you can see various chrome extension like that for example: Quick Notes

2. Disord Bot Intermediate

Description: The Discord Bot is a software application designed to enhance the functionality and user experience within the Discord platform. It provides automated features, moderation tools, utility functions, and interactive commands that can be used in Discord servers. The bot aims to streamline server management, provide entertainment, and facilitate engagement among server members.

Specification:

- Discord Integration: The bot should be integrated with the Discord platform to interact with servers and users.
- Command Handling: The bot should interpret commands from users and have a framework for creating and managing custom commands.
- Utility Functions: The bot should provide server administration commands and utility features like user and role management.
- Moderation Tools: The bot should have moderation features for enforcing server rules and handling moderation actions.
- Interactive Features: The bot can include interactive elements like polls, quizzes, or giveaways to engage server members.
- Integration with APIs and Services: The bot can integrate with external APIs or services for additional functionality.
- Event Handling: The bot should listen to Discord events and respond accordingly, such as message events or user join/leave events.

- Logging and Reporting: The bot should maintain logs of actions and generate reports for server administrators.
- Permissions and Security: The bot should have permission management and implement security measures to protect access.
- Scalability and Performance: The bot should be designed to handle multiple servers and interactions efficiently.
- Documentation and Support: The bot should have clear documentation and support channels for users.

Finance

1. Stock screener based on technicals with a frontend Beginner

Description: Create a platform that can show the candlestick graph of a stock for a period of 5 Years.

Specification:

- 1) The user can choose among the stocks that need to be displayed. For this purpose, you can use either US or Indian stocks. 2) The user can display some technical indicators on top of the graph of the particular stock chosen. At least 5 technical indicators need to be present for the same.

2. Paper trading stock simulator Intermediate

Description: Create a platform that can be used for paper trading i.e. trading using virtual money of either Indian or US stocks

Specification:

- 1) Use data from either the top 200 US stocks or the top 100 Indian stocks according to their market capitalisation. 2) It should allow the user to create a virtual wallet with virtual money to trade with during Intraday / Swing Trading.

3. Robo-advisory for Retirement Planning Intermediate

Description: Create a platform that can assist individuals in planning for their retirement.

Specification:

- 1) The system should consider factors like age, income, expenses, and life expectancy to provide personalized retirement savings and investment recommendations. 2) Additional Feature(Optional) The user can view the growth of their portfolio if they choose to invest in a particular asset class.

4. Personal finance management platform Intermediate

Description: The platform should provide various features to help users achieve their financial goals and make informed decisions about their money.

Specification:

- 1) The user can set financial goals that they want to achieve and the platform tells them the ways in which they can be achieved. 2) The users can keep track of their personal finances on the platform.

5. One-Stop solution for Traders Advanced

Description: The platform should provide a trader with all the necessary tools to practice and polish his trading skills.

Specification:

- 1) The platform should have a backtesting tab, a paper trading tab and also a custom indicator tab that is connected to the backtesting tab to test those indicators. 2) The user should be able to backtest his own strategy on any stock for a period of previous 5 years and also be able to paper trade on those stocks.

Game Development

1. Endless Runner Beginner

Description: Create a classic 2D endless runner game where players control a character that continuously runs and must navigate through obstacles, collect power-ups, and achieve high scores.

2. Retro Arcade Shooter Beginner

Description: Create a fast-paced 2D arcade shooter game inspired by classic games like Space Invaders. Players control a spaceship and must shoot down waves of enemy ships while avoiding their attacks.

3. Multiplayer Battle Arena Intermediate

Description: Build a fast-paced multiplayer 2D battle arena game where players compete against each other in real-time battles. Players can choose from a variety of unique characters, each with their own abilities and playstyles. The game could offer different game modes, team-based battles, and skill-based matchmaking to create a competitive and exciting experience.

4. Sci-Fi Racing Challenge Intermediate

Description: Develop a futuristic sci-fi 3D racing game where players compete in high-speed races across visually stunning tracks. The game could offer a variety of customizable vehicles with unique abilities, an array of power-ups and boosts. Players can strive to set records, unlock new tracks, and rise through the ranks.

5. 3D Puzzle Platformer: Time Manipulation Advanced

Description: Develop a puzzle platformer game where players manipulate time to solve intricate challenges. Players can rewind, fast-forward, or pause time to overcome obstacles, manipulate objects, and alter the environment. The game could feature mind-bending puzzles, visually stunning 3D worlds, and a captivating narrative centered around time manipulation.

6. Vehicular Combat Arena Advanced

Description: Create a multiplayer game where players engage in intense vehicular combat in a large arena. Players can customize and upgrade their vehicles, choose from different weapon loadouts, and strategize with their team to outmaneuver and destroy opponents. The game could offer a variety of game modes, destructible environments, and dynamic vehicle physics.

Material Design and Process Optimization

1. Determining the average life of a component Beginner

Description: Develop a platform that will take the average life and operating condition as input and will output the average time for which the component can be used

Specification:

- EDA, Data Visualization, ML, Python

2. Pivotal characteristic for achieving superconductivity Intermediate

Description: Performing simulations on various materials and using various algorithms to find the best-suited property for a material to achieve superconductivity.

Specification:

- Extraction of information about materials, simulations on materials, ml models

3. Ideal Alloy composition Intermediate

Description: Develop a platform that will tell the combination of metals to form an alloy suitable for a particular operation.

Specification:

- Extract data on various alloys and analyse how each metal contributes to the alloy's property.

4. Stress concentration analysis Intermediate

Description: Analysis of various factors affecting the component's life and using algorithms to find the stress concentration in the component and plotting a contour curve giving the stress analysis

Specification:

- Extracting data, Machine Learning, plotting libraries

Quantum Computing

1. Variational Quantum EigenSolver (VQE) Beginner

Description: A Quantum Computing program to estimate the energies of formation of Water molecules.

Specification:

- Quantum Chemistry, Variational Algorithms, Qiskit Nature

2. Quantum Compiler Challenge Beginner

Description: Design a Compiler for a Quantum Computer that allows to execute a set of gates on a given Hardware

Specification:

- Quantum Circuit Design. Linear Algebra, Optimization

3. Quantum KNN Classifier Intermediate

Description: A Quantum Program to classify MNIST Digits based on KNN Classification

Specification:

- Quantum Machine Learning, KNN Algorithm

4. Travelling Salesman Problem Advanced

Description: Execute the Travelling Salesman Problem on a Quantum Computer.

Specification:

- Computer Optimization Techniques involving Quantum Algorithms

Web development

1. CP Portal Beginner

Description: Create a responsive centralised Web Application that users can use to access various competitive programming websites from a single place.

Specification:

- The application must allow the user to:
- Receive updates on upcoming contests.
- Add problems to a To-Do list.
- Select and save favourite problems.
- Get random problem recommendations based on the problem rating and topic constraints.

2. Quiz Website Beginner

Description: Create Responsive Quiz taking Platform with required specifications

Specification:

- Develop a web-based quiz-taking platform that allows users to participate in various quizzes on different topics.
- The website should have a user registration and login system to track individual users and their progress.
- Users should be able to browse and select from a list of available quizzes categorized by topics, difficulty levels, or other relevant criteria.
- Each quiz should consist of a set of multiple-choice questions with a defined time limit for completion.
- The platform should provide immediate feedback to users, displaying their score and correct/incorrect answers on quiz completion.
- Users should be able to view their quiz history, track their progress, and compare their scores with other participants.
- Provide an admin panel for managing quizzes, creating new quizzes, and monitoring user activity.
- Bonus points for making the exam proctored and adding face detection to prevent cheating.

3. Night Canteen Ordering Portal Beginner

Description: Create a responsive website for Night Canteen On IITI

Specification:

- Create a user interface for both admin and users.
- Enable users to access and view the updated list of available food items in the night canteen, which is maintained and updated by administrators.
- Allow users to conveniently place their food orders through the website, providing them with a streamlined ordering process.
- Develop a website that effectively addresses key challenges faced in the night canteen, aiming to provide practical solutions and enhance the overall experience for users.
- Bonus Points: You can expand this system for Tapri IITIans ki, Bakeology, Noor, Aladdin etc.

4. Github Explorer Beginner

Description: GitHub Explorer is a web-based application that allows users to explore and discover open-source projects hosted on GitHub. It provides a user-friendly interface for searching, browsing, and accessing repositories based on various criteria such as programming language, popularity, stars, forks, and more. The application aims to simplify the process of finding relevant projects for developers, researchers, and anyone interested in exploring the vast collection of open-source software available on GitHub.

Specification:

- The application should integrate with the GitHub API to fetch repository information, user data, and activity updates.
- The application should be responsive and accessible across different devices, including desktops, tablets, and mobile devices.
- User Authentication.
- Users can view the recent activity in a repository, such as commits, pull requests, and issues.
- When a user selects a repository from the search results, detailed information about the repository should be displayed.
- Users can search for repositories based on keywords, programming language, and other relevant filters.
- And many more features can be added.
- Bonus Points: Implement a Recommendation System based on their recent search history or their liking in particular fields.

5. Counselling Appointment System Beginner

Description: Create a responsive Web Application for booking appointments with the counsellor at our college.

Specification:

- The application must allow the user to:
  1) View a calendar of each counsellor depicting their available time slots.
  2) Select their desired counsellor's available time slot from the counsellor's calendar.
  3) After sending an appointment request, the time slot for that counsellor should freeze for other users.
  4) Freeze colour should change upon acceptance of the appointment. (e.g. initially green depicting available slot, after sending request yellow, after acceptance red depicting booked slot for other users)
  5) Cancel appointments.
  6) View each counsellor's profiles
- The application must allow the counsellor to:
  1) Manage its appointments, such as cancelling, accepting and rescheduling them.
  2) Add, delete and modify their available time slots on their calendar
  3) Manage their profiles

6. Alumni Portal Intermediate

Description: Create a responsive Web Application for IITI Alumni. The portal should connect all the alumni and serve as the current Website for the Alumni Cell of IIT Indore (which is currently run by third parties). The portal could act as LinkedIn for the IIT Indore Community.

Specification:

- The application allows the Alumni Cell Members (Admins) to:
  1) Post all upcoming events organized by the cell and ask the alumni to register for them.
  2) View the count of those registered and those who only viewed the page.
  3) Send mail to all/specific alumni regarding various events.
  4) Send automated birthday wish Emails to each alumnus on their birthday.
  5) Authenticate users who have registered on the portal. On successful authentication, the users should be able to access all the features on the website.
  6) Upload pictures of all the events conducted.
- The application allows the Alumnus to:
  1) Create a profile on the portal with their contact and job details.
  2) On successful login, the alumnus can view their friends' profiles and chat with them.
  3) Post job opportunities for other alumni and also be able to apply for the same.
  4) View the number of alumni by their count as per their location, company, and year of graduation.
- The application allows external viewers to:
  1) Only view the pages of the portal (cannot interact with any of the registered users)

7. Event Ticketing Platform Intermediate

Description: Portal where Events can be posted by different organisations, and tickets can be bought by the users

Specification:

- The application allows the Organizers(Admins) to:
  1) Post upcoming events and their description
  2) add tickets for its sale.
  3) View purchase tickets by the users.
  4) scan the QR code tickets sent to users on the day of the event. to allow them entry.
  5) create coupon codes, which can be used by users during their purchases.
- The application allows users to: 1) Select and view the events posted by the Organizers
  2) Sign-up/log in and buys tickets for any desired event
  3) Upon purchase of any ticket, its name must be added to that event's database, which can be viewed by the organizers.
  4) A confirmation email must be sent after the successful purchase of the ticket, along with a unique QR code.
  5) Create, View, and modify its profile on the website.

8. User Auth Microservice Intermediate

Description: The User Authentication Microservice is a standalone component that handles user authentication and authorization functionalities for a larger system or application. It provides a secure and scalable solution for managing user accounts, authentication, and access control. The microservice is responsible for user registration, login, password management, token generation, and other related authentication operations.

Specification:

- Users can create new accounts by providing necessary information such as username, email address, and password.
- The microservice should validate and store user registration data securely.
- Optional registration features such as email verification or a captcha can be implemented to enhance security.
- Registered users can authenticate themselves by providing their credentials (username/email and password) to the microservice.
- The microservice should verify the credentials and issue an authentication token upon successful login.
- Upon successful authentication, the microservice should generate an access token that can be used for subsequent requests.
- The microservice should securely store and manage tokens and provide mechanisms for token validation and expiration.
- The microservice should provide APIs or integration points for other services or applications to interact with it for authentication and user-related operations.

- Proper authentication and authorization protocols (e.g., OAuth, JWT) can be implemented to facilitate secure communication between the microservice and other components.
- The microservice should log relevant authentication events and activities for monitoring, auditing, and troubleshooting purposes.

9. Algorithm Visualiser Intermediate

Description: Develop a website dedicated to algorithm visualization, which enables users to visually comprehend different algorithms such as sorting algorithms (e.g., bubble sort, insertion sort) and graph algorithms (e.g., DFS, BFS).

Specification:

- Create an intuitive and user-friendly interface that allows users to easily navigate and interact with the website.
- Provide a wide range of algorithms for visualization, including sorting algorithms (e.g., bubble sort, insertion sort, quicksort) and graph algorithms (e.g., DFS, BFS, Dijkstra's algorithm).
- Include controls to start, pause, and reset the visualization, as well as step-by-step visualization options for a more detailed understanding of the algorithm's execution.
- Provide accompanying code snippets and concise explanations for each algorithm, giving users a better understanding of the underlying logic and implementation details.

10. Slack Clone Intermediate

Description: Create a Slack clone

Specification:

- Implement a user registration system that allows users to create accounts and authenticate themselves securely.
- Enable users to create public or private channels for group communication and one-on-one direct messaging for private conversations.
- Allow users to upload and share files, such as documents, images, and videos, within channels and direct messages.
- Implement notifications for new messages, mentions, and other relevant activities to keep users informed even when they are not actively using the website.
- Implement a search feature that enables users to search for messages, files, channels and users to quickly find relevant information.
- Apart from that, try to implement all relevant features of Slack

11. Video Conferencing Advanced

Description: The Video Conferencing Website is a web-based platform that enables users to communicate, collaborate, and conduct video conferences in real time. It provides a virtual meeting space where participants can join meetings, share audio and video, exchange messages, and collaborate on documents or presentations. The website aims to provide a seamless and user-friendly experience for remote communication and collaboration.

Specification:

- User Registration and Authentication: Users can create accounts and log in to the website using their email addresses, social media accounts, or other authentication mechanisms. The website should handle user authentication securely and protect user privacy.
- Meeting Management: Users can create new meetings and schedule them for specific dates and times. The website should provide options for inviting participants, setting meeting agendas, and managing meeting details.
- Real-Time Video and Audio Conferencing: The website should support real-time video and audio communication among meeting participants. Participants should have the ability to enable/disable their video and audio streams as needed. Features like screen sharing and virtual backgrounds can be included to enhance the conferencing experience.
- Chat and Messaging: Participants should be able to send text messages in real time during the meeting. Private Messaging options can be included for one-on-one communication between participants.
- Collaboration Tools: The website can provide collaboration tools such as document sharing, whiteboarding, and file sharing. Participants should be able to work together on shared documents, make annotations, and collaborate on projects.
- Meeting Recording and Playback: The website can offer the ability to record meetings for future reference or for participants who were unable to attend. Recorded meetings should be securely stored and easily accessible for playback.
- Meeting Invitations and Reminders: The website should provide options for sending meeting invitations and reminders to participants via email or notifications. Calendar integration (e.g., Google Calendar) can be implemented to simplify scheduling and reminders.
- Security and Privacy: The website should prioritize security and implement encryption mechanisms to protect user data and communication. Privacy controls can be included to allow participants to manage their preferences for sharing personal information.
- User Interface and User Experience: The website should have an intuitive and user-friendly interface that makes it easy for participants to navigate and use the conferencing features. The user interface should be responsive and accessible across different devices and screen sizes.
- Moderation and Administration: The website can include features for meeting moderation, allowing designated moderators to manage participants, control meeting settings, and handle disruptive behaviour if necessary.
- Integration and Compatibility: The website should be compatible with popular web browsers and operating systems to ensure broad accessibility. Integration with other tools and platforms, such as calendar systems or project management tools, can be considered to enhance productivity.

12. File Transfering Advanced

Description: The File Transfer Application is a software solution that allows users to securely and efficiently transfer files between different devices or across a network. The application aims to simplify the process of sharing files, whether it is between users on the same network or over the internet. It provides a user-friendly interface for selecting files, initiating transfers, and monitoring the progress of file transfers.

Specification:

- The application should have an intuitive and user-friendly interface that allows users to easily navigate and interact with the file transfer functionalities.
- Users should be able to select one or multiple files or directories for transfer.
- The application should handle both small and large files efficiently, ensuring reliable and fast transfers.
- Support for resumable transfers and the ability to handle interrupted or failed transfers should be included.
- The application should support multiple transfer protocols, such as FTP, SFTP, HTTP, or a custom protocol.
- The choice of protocols can depend on the specific requirements and security considerations of the project.
- The application should prioritize security and implement encryption mechanisms to protect file transfers.
- Secure protocols (such as SFTP or HTTPS) or encryption algorithms can be used to ensure the confidentiality and integrity of transferred files.
- The application should be compatible with multiple operating systems and devices, including Windows, macOS, Linux, and mobile platforms, if applicable.
- Users should be able to track the progress of file transfers, including transfer speed, estimated time remaining, and completed percentage.

Cybersecurity

1. Cyberchef clone Intermediate

Description: The CyberChef Clone is a web-based application that replicates the functionality of the popular CyberChef tool. It provides users with a platform to perform various cryptographic and data manipulation operations in a user-friendly manner. The application aims to simplify the process of encoding, decoding, analyzing, and transforming data for cybersecurity professionals, developers, and anyone working with data manipulation tasks.

Specification:

- The application should have a clean and intuitive user interface that allows users to easily interact with the different operations and functionalities.
- Output results should be displayed in a readable format and presented to the user in an organized manner.
- Users should be able to perform various encoding and decoding operations, including but not limited to Base64, URL encoding, HTML encoding, hexadecimal, binary, and more.
- The application should support bidirectional conversion, allowing users to encode and decode data back and forth.
- The application should support common cryptographic operations such as hashing (MD5, SHA-1, SHA-256, etc.), encryption (AES, RSA, etc.), and decryption.
- Users should be able to input keys or passwords for cryptographic operations when required.
- The application should support file upload functionality, allowing users to process and manipulate data from uploaded files.
- The application should be responsive and accessible across different devices, including desktops, tablets, and mobile devices.

2. Chall Status Ping Advanced

Description: The SSH Server Ping and Challenge Status Checker is a tool designed for cybersecurity purposes. It allows users to ping an SSH server and retrieve the status of various challenges within a cybersecurity competition. The tool helps assess the availability and responsiveness of the SSH server while providing an overview of the status of individual challenges.

Specification:

- SSH Server Connection: The tool should establish an SSH connection with the target server using the provided credentials (IP address, username, and password/private key).

- Ping Functionality: The tool should ping the SSH server to check its availability and response time. It should display the ping statistics, including packet loss percentage, round-trip time (RTT), and average response time.
- Challenge Status Retrieval: The tool should retrieve the status of various challenges hosted on the SSH server. It can use SSH commands or protocols (e.g., SSH sessions, SCP) to interact with the server and fetch the challenge status.
- Challenge Status Display: The tool should present the challenge status in a clear and organized manner. It can display a list of challenges along with their respective statuses, such as "solved," "unsolved," "in progress," or "locked."
- Challenge Details: The tool can provide additional details for each challenge, such as the challenge description, point value, and any relevant hints or instructions.
- User-Friendly Interface: The tool should have a user-friendly command-line interface or graphical user interface (GUI) that simplifies the process of pinging the SSH server and checking challenge statuses. Clear instructions and prompts should be provided to guide users through the tool's functionalities.
- Error Handling: The tool should gracefully handle potential errors, such as invalid credentials, connection failures, or challenge retrieval issues. It should display meaningful error messages and suggest troubleshooting or resolving the issues.
- Logging and Reporting: The tool can log the ping results, challenge statuses, and any errors encountered during the process. It can generate reports summarizing the overall status of the SSH server and challenges for further analysis or documentation.
- Security Considerations: The tool should ensure the secure handling of user credentials and protect sensitive information during the SSH connection process. It should follow best practices for secure coding and implement necessary safeguards to prevent unauthorized access or data leakage.
- Compatibility: The tool should be compatible with operating systems like Windows, macOS, and Linux. It should support SSH connections and commands across various platforms.
- Documentation and Help: The tool should provide comprehensive documentation, including installation instructions, usage examples, and troubleshooting guidance. Help menus or online resources can be included to assist users in navigating the tool's functionalities.

3. CTFd Framework Advanced

Description: The Capture The Flag (CTF) Framework, commonly known as CTFd, is a web-based platform designed to host and manage Capture The Flag competitions. It provides an environment for participants to solve various challenges, such as cryptography, reverse engineering, web exploitation, and more, to earn points and compete with other players. CTFd aims to offer a robust and customizable framework for organizing, scoring, and tracking CTF competitions.

Specification:

- User Registration and Authentication: CTFd should support user registration and authentication to allow participants to create accounts and access the plat-

form. Different authentication methods, such as username/password, email verification, and OAuth integration, can be supported.

- Challenge Management: CTFd should provide a user-friendly interface for administrators to create, manage, and categorize participant challenges. Challenges can include various categories, difficulty levels, descriptions, and attachments for participants to solve.
- Scoring System: CTFd should implement a scoring system to track and calculate points earned by participants for solving challenges. Customizable scoring rules, such as different point values for each challenge or dynamic scoring based on completion time, can be supported.
- Team Management: CTFd should allow participants to form or join teams to compete collectively. Team management features, such as creating teams, inviting members, or managing team details, should be included.
- Real-Time Scoreboard: CTFd should display a real-time scoreboard that updates participant scores and rankings during the competition. The scoreboard can be publicly accessible or restricted to authorized users.
- Hints and Progress Tracking: CTFd can offer hints or guidance for participants to help them progress through challenges. Progress tracking features can be included, such as marking completed challenges or tracking individual participant progress.
- Secure Challenge Deployment: CTFd should provide a secure environment for deploying challenges and protecting participants' solutions and submissions. Measures like sandboxing or containerization can be implemented to isolate challenges and prevent exploitation of the underlying system.
- Admin Dashboard: CTFd should have an admin dashboard that allows administrators to manage competitions, view participant activity, and perform necessary configurations. Admin features may include challenge creation, user management, scoreboard customization, and result analysis.
- Documentation and Support: CTFd should provide comprehensive documentation on how to set up and configure the framework. Support channels, such as a dedicated forum or community, can be established to assist users and address any issues or questions.
- Customization and Extensions: CTFd should offer customization options to tailor the platform's appearance, scoring system, and other functionalities to meet specific competition requirements. Extension mechanisms, such as plugin support or API integration, can be provided to allow for additional features or integration with external services.
- Security and Privacy: CTFd should prioritize security and implement measures to protect participant data, challenge solutions, and the platform against potential vulnerabilities or attacks.

On-site Concrete Technology

1. Crack detection in concrete surface Intermediate

Description: The objective of this PS is to develop a machine-learning solution for crack detection in concrete structures. The goal is to design an accurate and efficient system that can automatically identify and classify cracks in concrete using computer vision techniques. The system should be capable of analyzing images or videos of concrete surfaces and accurately detecting the presence of cracks, their sizes, and their locations..

Specification:

- Dataset Acquisition: Collect a diverse and representative dataset of concrete images or videos, encompassing various types of cracks, lighting conditions, and surface textures. The dataset should be appropriately annotated with ground truth labels indicating crack locations and categories.
- Preprocessing and Augmentation: Apply preprocessing techniques such as image normalization, denoising, and contrast enhancement to enhance the quality and consistency of the dataset. Augment the dataset by employing transformations like rotation, scaling, and flipping to improve model generalization.
- Crack Detection and Localization: Develop a machine learning model capable of accurately detecting and localizing cracks in concrete images or frames. The model should leverage computer vision algorithms, such as convolutional neural networks (CNNs) or deep learning architectures, to identify crack patterns and distinguish them from non-crack regions.
- Accuracy and Robustness: Aim for high accuracy in crack detection and classification, minimizing false positives and false negatives. Address challenges such as variations in lighting conditions, surface textures, and crack sizes to build a robust system that performs consistently across different scenarios.
- User Interface and Integration: Design a user-friendly interface that allows users, such as engineers or inspectors, to easily input concrete images or videos and obtain crack detection results. Integrate the solution into existing inspection workflows or maintenance management systems, ensuring seamless adoption and usability.
- Model Evaluation and Validation: Establish appropriate evaluation metrics and methodologies to assess the performance of the crack detection and classification models. Conduct rigorous testing and validation using both the training dataset and additional test datasets to ensure reliability and generalization.
- Deployment and Scalability: Develop a scalable and deployable system that can handle large-scale crack detection tasks across multiple concrete structures. Consider cloud-based solutions, distributed computing, or parallel processing techniques to efficiently process and analyze extensive datasets.