GDSC DJSCE

SYNAPSE

HACKNICHE 2.0

EXPERIENCE THE TECHSAKURA

TITLE SPONSOR

POWERED BY

SILVER SPONSOR

Problem Statements

SWAG PARTNER :

GAMING PARTNER :

GuidePost Consult

GDSC DJSCE

HACKNICHE 2.0

SYNAPSE

WEB/APP

GDSC DJSCE

HACKNICHE 2.0

SYNAPSE

PS 1-Portfolio Management

Managing an investment portfolio effectively requires a balance between risk and return. In the evolving landscape of financial technology, there is a growing demand for automated solutions that can provide users with intelligent portfolio management services.

Historical Market Data Integration:

- Ingest and process historical market data from various financial instruments and markets.
- Ensure the tool accommodates a wide range of data frequencies (e.g., tick, minute, daily) and time periods.
- Display this data on an attractive dashboard where insights can be gleaned intuitively.

Data Visualization and Stock Graphs:

- Integrate interactive stock graphs for visualizing the historical performance of individual securities and the overall portfolio.
- Include features for comparing portfolio performance against benchmark indices.

Real-time market alerts:

- Display articles related to current financial scenarios and portfolios for a relevant news feed.
- Provide tools for users to set risk parameters, such as stop-loss levels, position sizing etc. Provides notifications for the same.

Dynamic Portfolio Rebalancing:

- Develop algorithms for creating diversified portfolios. Ensure the inclusion of various asset classes, such as stocks, bonds, and possibly alternative investments. Enable automatic rebalancing of portfolios to maintain the desired asset allocation over time.
- Implement algorithms that trigger rebalancing based on market conditions, economic indicators, or predefined rules.

GDSC DJSCE

HACKNICHE 2.0

SYNAPSE

PS 1-Portfolio Management

Bonus:

- Incorporate principles from behavioral finance to help users make more rational decisions during periods of market volatility.
- This could include nudges, educational content, or simulations to illustrate the impact of emotional decision-making.

GDSC DJSCE

HACKNICHE 2.0

SYNAPSE

PS 2-Food Truck Application

You have been hired by a food truck company to build a web/app platform for both administration and client-side operations. They want an admin side for the management of the actual food truck management by the business owners. Also, they want a client-side application for the consumers. The actual Problem Statement entails:

Client side:

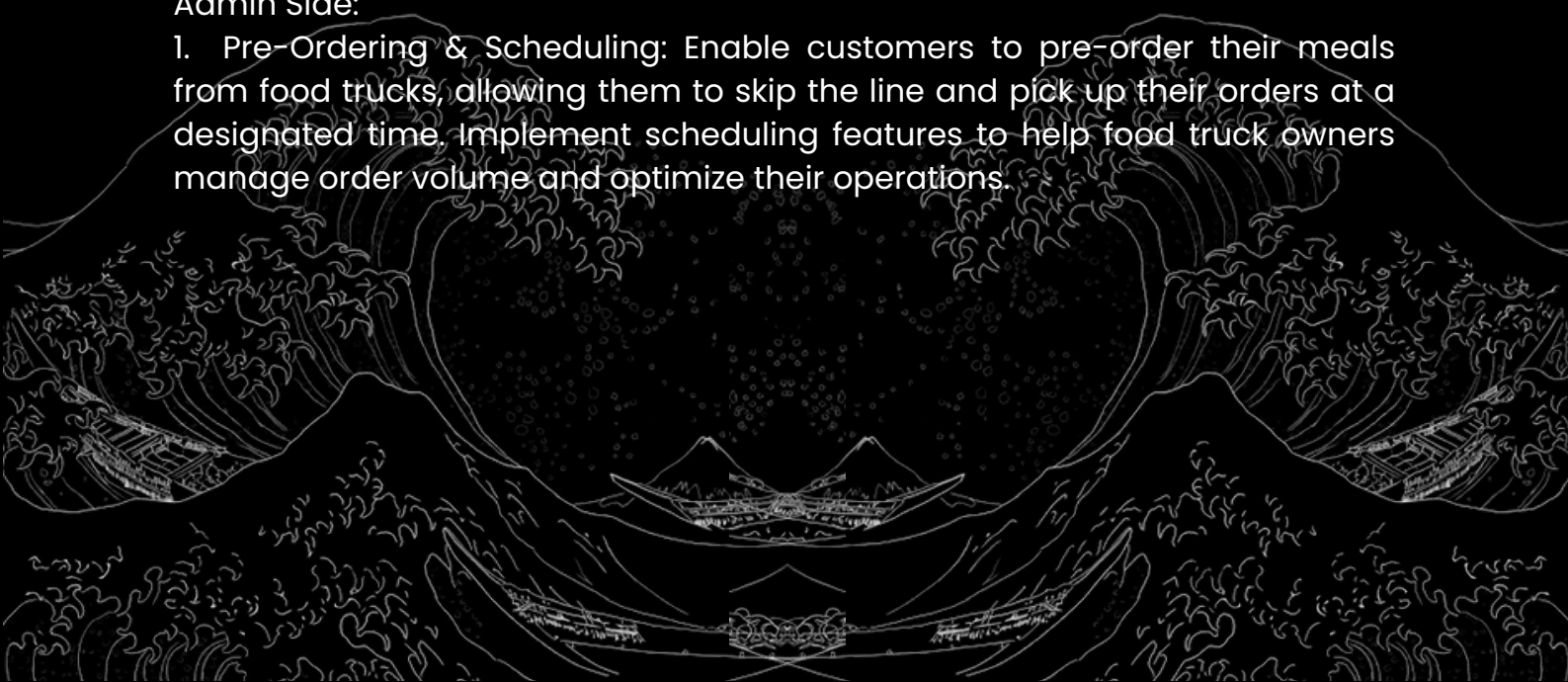
1. Innovative menu display: The traditional list format of displaying menus is dreary and dull now. Come up with innovative ways to display the menu items using animations, food pictures, carousels, search functionalities and whatever else your creativity can imagine. At the same time, keep in mind that display of information such as Jain, Veg, Non-Veg, Price etc. should not be compromised.

2. Location & schedule display: When the food truck is changing locations, the user should get a notification from the food truck stating that they have now changed locations, and they can avail their services from their new site now. Notifications should also reflect upon change in menus, daily special items, and special events, promotional deals.

3. Social Media Integration: Enable users to share their food truck experiences on social media platforms directly from the app. Integration with social media APIs can also allow food truck owners to showcase their latest offerings and engage with their audience.

Admin Side:

1. Pre-Ordering & Scheduling: Enable customers to pre-order their meals from food trucks, allowing them to skip the line and pick up their orders at a designated time. Implement scheduling features to help food truck owners manage order volume and optimize their operations.



GDSC DJSCE

HACKNICHE 2.0

SYNAPSE

PS 2-Food Truck Application

2. Rating & Reviews: Implement a rating and review system where customers can provide feedback on their food truck experience. This helps maintain quality standards and build trust among potential customers.

3. Order Management Dashboard: Develop a dashboard for food truck owners to view and manage incoming orders, track inventory levels, and communicate with customers. Keep track of customer density based on location and other features. This should help in optimal route planning for the truck, whilst also ensuring proper waste disposal on road.

4. Inventory management: Come up with innovative inventory management solutions for all the necessary ingredients, foods, food equipment, cutlery, plates, truck fuel and all other items that may be required in a food truck.

BONUS:

Plain text is plain boring. Add
1. multilingual support,
2. OCR, and
3. voice input functionalities

in appropriate places for brownie points.

GDSC DJSCE

HACKNICHE 2.0

SYNAPSE

GDSC DJSCE

HACKNICHE 2.0

SYNAPSE

PS 3-Function Code from Text Prompts

Challenge:
Generate Function Code from Text Prompts

Context:
Nowadays automating code generation becomes increasingly valuable. This hackathon
challenges you to build a tool that creates function code based on user-provided input,
enabling rapid prototyping and exploration of data processing algorithms.

Objectives:
Code Generation from Text Prompts: The tool should translate a natural language prompt describing the desired functionality into efficient, working code.
Structured Input and Output: Users can provide an input JSON object, an output
JSON object, and a database schema (SQL or MongoDB) to guide the code generation process.

Real-time testing and validation:
The tool should be able to test and validate the code that it generates and report and optimise the performance statistics of the code.
Clean UI and Backend: Design a user-friendly interface and robust backend API
using Node.js or Python to facilitate interaction and code generation.
Pseudocode Generation: Utilize LLMs to create intermediate pseudocode steps
as stepping stones to final code generation.
LLM-Assisted Development: Leverage open-source LLMs and APIs to generate and debug code effectively.
Efficient Prompt Creation: Focus on crafting concise and effective text prompts
that accurately convey the desired function's logic and behavior.

GDSC DJSCE

HACKNICHE 2.0

SYNAPSE

PS 3-Function Code from Text Prompts

Evaluation Criteria:
Code Functionality: Correctness, efficiency, and adherence to provided requirements.
Usability and Design: Clarity, intuitiveness, and user-friendliness of the IJI.
Technical Implementation: Code quality, adherence to best practices, and effective LLM utilization.
Prompt Creation Skill: Ability to formulate precise and informative prompts for LLM-based code generation.

Bonus:
Integration with additional features like code explanation, performance analysis, or version control.

Resources:
•Gemini Documentation
Open-source LLMs
Recommended IJI Frameworks:
React
Recommended Backend programming Languages:
Node.js
Python
GoLang
Additional Notes:
Open-source libraries and frameworks are strongly encouraged.
Creativity and innovation in prompt engineering are key to success.
All code and documentation must be the team's original work.
Have fun and showcase your AI/ML expertise!

GDSC DJSCE

HACKNICHE 2.0

SYNAPSE

PS 4-Cafe Analytics and AI-Driven Insights

Imagine a bustling coffee shop eager to elevate its game in the highly competitive market. To achieve this, the problem statement is to harness the power of data analytics and AI. The coffee shop wishes to revolutionize its operations, decision-making processes, and customer engagement strategies. Keep high focus on documentation, git version control, good coding practices, making a proper ReadMe and so on.

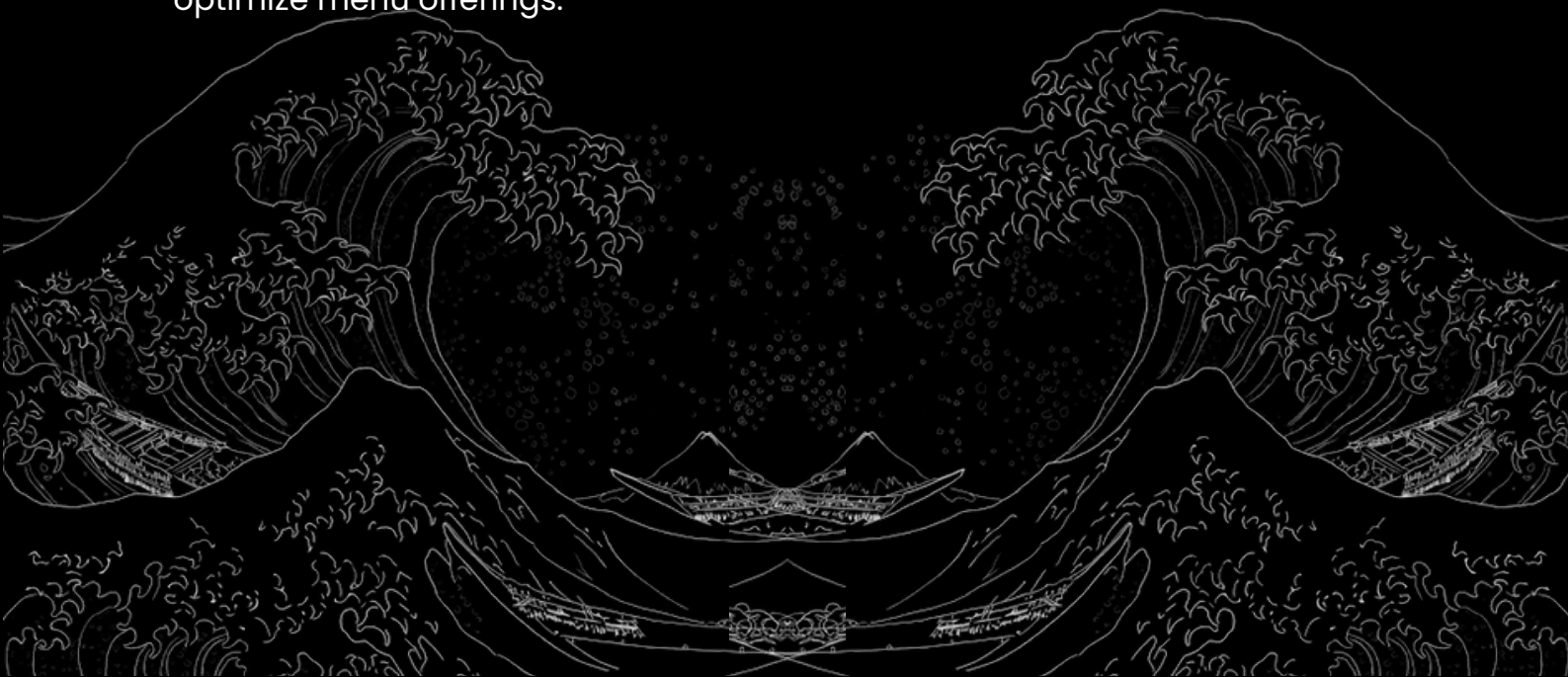
Core objectives:

1. **Insight Extraction: Analyze Customer Reviews and Competitor Data:**

- Utilize web scraping techniques to extract user reviews from platforms like Zomato and Swiggy for the Target Restaurant (Ettarra Coffee). Also consider factors such as cuisine and other searchables.
- Analyze customer reviews to understand the strengths and weaknesses of the coffee shop and its competitors.
- Compare and contrast competitors' (Similar restaurants in the region) data to identify areas for improvement and strategic opportunities. Example: If you go to Zomato and search biryanis in the search bar then the top 10 competitors of that region are shown based on the relevance. Now with this competitor basket, curate the summary of how their business is doing.

2.Market Basket Analysis: Understand Customer Buying Patterns

- Analyze transaction data to identify rules/patterns in customer purchasing behavior.
- Determine associations between products using market basket analysis to optimize menu offerings.



GDSC DJSCE

HACKNICHE 2.0

SYNAPSE

PS 4-Cafe Analytics and AI-Driven Insights

3.AI Recommender System: Enhance Customer Experience

- Develop an AI-based recommender system for menu items based on customer preferences.
- Utilize the dataset provided to do so.

4.Analytics Dashboard Development: Provide Comprehensive Business Insights

- Create a dynamic dashboard that consolidates insights from customer reviews, competitor analysis, and market basket analysis.
- Visualize key performance indicators, customer sentiment trends, and sales patterns for quick decision-making.

5.Sales Analysis:

- Examine sales data to pinpoint intervals with reduced transaction activity, such as during extended weekends when patrons usually travel to other cities, and communicate these findings to the management. This information can then be utilized to launch targeted seasonal promotions.
- Also peruse purchases data and see how those insights can be used in these targeted seasonal promotions.

Bonus Challenge:

Use Open-Source LLMS to enhance the analytical capabilities: Utilize these LLMS to curate analysis of the data, enabling detailed analysis and pattern recognition. Uncover hidden insights and gain a competitive edge in decision-making.

GDSC DJSCE

HACKNICHE 2.0

SYNAPSE

GDSC DJSCE

HACKNICHE 2.0

SYNAPSE

PS 5 - Real Time Docker Container Monitoring Tool

Develop a Real-Time Docker Container Monitoring Tool Context:
As containerized applications are gaining traction, efficient monitoring is crucial. This hackathon challenges you to create a real-time Docker container monitoring tool that empowers users to gain invaluable insights into their containerized environments.

Objectives:

- Real-Time Monitoring:

The tool should display key container metrics, including CPU utilization, RAM usage, network traffic, and error logs, in real-time, providing a dynamic view of container health and performance.

- Comprehensive Information: Gather and present metrics for essential aspects like container restarts, resource consumption trends, and error frequency, enabling proactive problem identification and resolution.
- User-Friendly Interface: Design a clean and intuitive UI using either Node.js, Python, or GoLang, ensuring seamless interaction and data visualization for users of all levels.
- Docker API Integration: Leverage the official Docker APIs to extract container metrics effectively.
- Scalability and Reliability: Construct the tool to handle large numbers of containers and diverse monitoring requirements, ensuring robustness and stability under load.
- Alerting and Automation (Optional): Consider incorporating functionality for customizable alerts and automated actions based on pre-defined thresholds or error conditions, enhancing proactive management. Evaluation Criteria:
- Effectiveness of Monitoring: Accuracy, comprehensiveness, and real-time nature of displayed metrics.
- Usability and Design: Clarity, intuitiveness, and user-friendliness of the UI.
- Technical Implementation: Code quality, adherence to best practices, and use of Docker APIs.
- Scalability and Reliability: Demonstration of ability to handle diverse monitoring requirements and potential load.

GDSC DJSCE

HACKNICHE 2.0

SYNAPSE

PS-5 Real-Time Docker Container Monitoring Tool

Optional (Bonus): Implementation of alerting and automation features, adding to the tool's value proposition. Resources:

- Official Docker Documentation: https://docs.docker.com/
- Docker API Reference: https://docs.docker.com/engine/api/v1.44/
- Recommended UI Frameworks: ○ React ○ HTML, CSS, Javascript
- Recommended Backend programming Languages: ○ Node.js ○ Python ○ GoLang

Additional Notes:

- Open-source libraries and frameworks are highly encouraged.
- Feel free to be creative and add innovative features to enhance the tool's functionality.
- All code and documentation must be the team's original work.
- Have fun and showcase your talent!