Code Samples and Explanations

Arshak Shan Shajahan

Introduction

- Projects and Code Samples
- Github Link to each project is at the bottom of the second slide for each project.
- Readme.md Provided to all Github Projects

Contact:

- ashajaha@asu.edu
- **Visit my portfolio (https://arshakshan.com)**
- LinkedIn
- GitHub (https://github.com/arshakshan)

E-Commerce Website (Full Stack Web App)

Description:

Developed a high-performance e-commerce website using Next.js, delivering a responsive and intuitive shopping platform. Implemented Stripe for secure transactions, integrated Payload CMS for backend management, and optimized for SEO.

Key Technologies:
Next.js 14, Stripe API, Payload CMS, React, Node.js, Express, SEO optimization techniques.

Key Features:

- Dynamic cart system
- Secure authentication workflows
- SEO optimized content
- Stripe Payment Integration
- Streamlined checkout process

E-Commerce Website (Full Stack Web App)

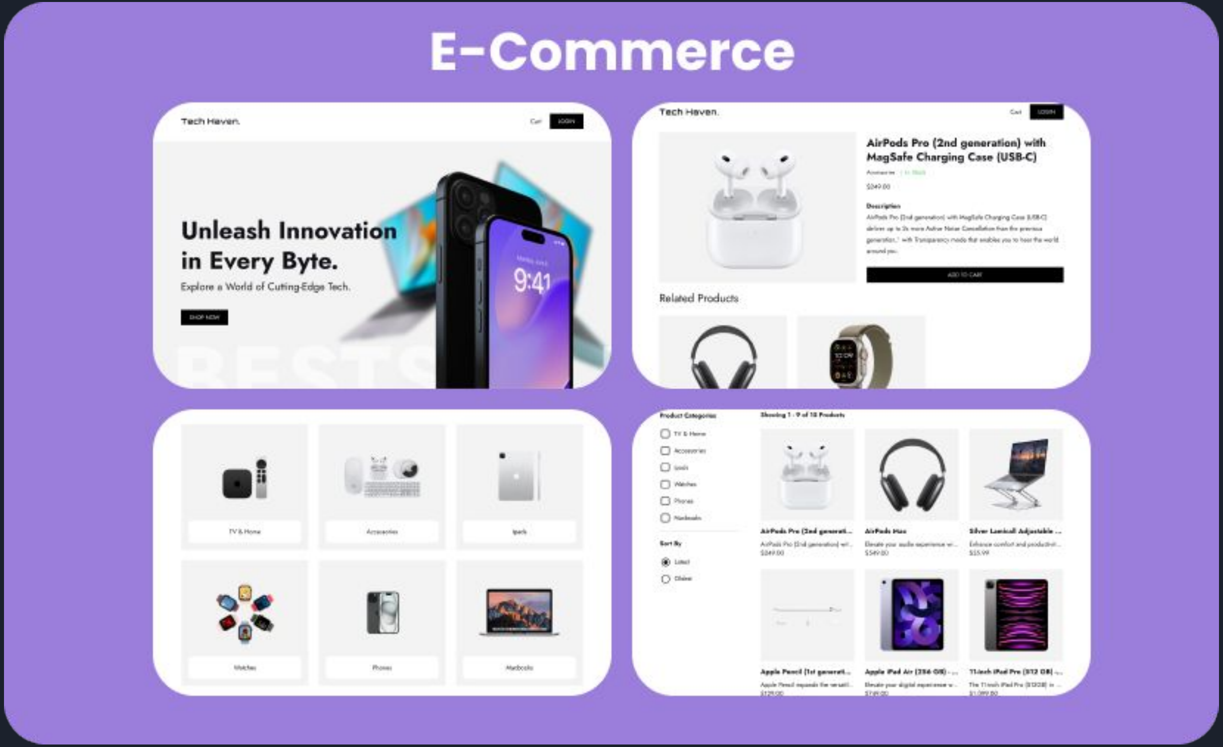
Technical Implementations:

- Dynamic Cart System: Implemented seamless addition/removal of products, real-time price calculations, and memory feature for returning users.
- Secure Authentication: Developed secure account creation, login, and password recovery workflows to enhance customer trust and privacy.
- Payload CMS Integration: Integrated Payload CMS for robust backend management, offering administrators an extensive admin panel for managing product listings, orders, and user data.
- SEO Optimization: Leveraged SEO best practices within Payload CMS to increase organic traffic and improve online visibility.
- Checkout Process: Streamlined with intuitive interface design and clear, step-by-step guidance, reducing cart abandonment rates and improving conversion metrics.

Link: GitHub (https://github.com/arshakshan/ecommerce)

Website: https://ecommerce-b2cc06c.payloadcms.app

E-Commerce Website (Full Stack Web App)



E-Commerce

Unleash Innovation in Every Byte.

Explore a World of Cutting-Edge Tech.

AirPods Pro (2nd generation) with MagSafe Charging Case (USB-C)

$249.00

Description

AirPods Pro (2nd generation) with MagSafe Charging Case (USB-C) delivers up to 2x more Active Noise Cancellation than the previous generation,¹ with Transparency mode that enables you to hear the world around you.

Related Products

Product Categories

□ TV & Home
□ Accessories
□ Ipads
□ Watches
□ Phones
□ Macbooks

Sort By

● Latest
○ Oldest

Showing 1 - 9 of 10 Products

AirPods Pro (2nd generati... AirPods Pro (2nd generation) wi... $249.00	AirPods Max Elevate your audio experience wi... $549.00	Silver Lamicall Adjustable ... Enhance comfort and producti... $15.99
Apple Pencil (1st generati... Apple Pencil expands the versatil... $129.00	Apple iPad Air (256 GB) - ... Elevate your digital experience w... $749.00	11-inch iPad Pro (512 GB) -... The 11-inch iPad Pro (512GB) is ... $1,099.00

Food Runner Android App

Description:

Engineered a food delivery app in Kotlin with features for seamless user interactions, robust review mechanisms, and efficient data management using JSON and SQLite.

Key Technologies:
Android Development, Kotlin, Postman API, RoomDB.

Key Features:

- User-friendly interface with Kotlin + XML
- Robust review feature
- Place Order, Cancel Order, Favorite Restaurants
- Efficient data management with SQLite and RoomDB
- Server interactions with REST APIs

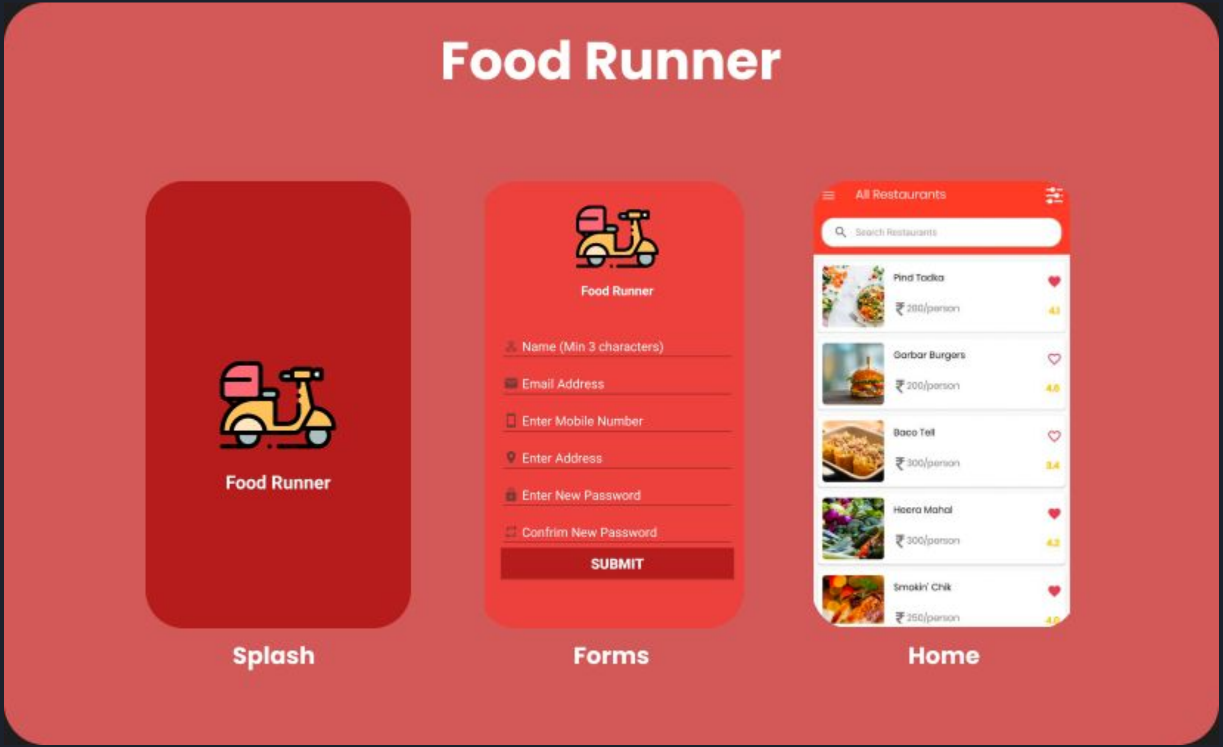
Food Runner Android App

Technical Implementations:

- User Interactions: Facilitated seamless interactions for users to explore menus, place orders, or cancel them with just a few taps, ensuring a pleasant user experience.
- Review Mechanism: Integrated a robust review feature allowing users to share their experiences, ratings, and feedback, fostering a community-driven environment.
- Data Management: Employed JSON for efficient server interactions and SQLite for local data storage, ensuring faster loading times and a smoother app experience.
- Development in Kotlin: Utilized Kotlin for robust, maintainable, and optimized app performance, ensuring reliability and security.

Links: Github (https://github.com/arshakshan/FoodRunner)

Food Runner Android App



Name (Min 3 characters)

Email Address

Enter Mobile Number

Enter Address

Enter New Password

Confrim New Password

SUBMIT

Pind Tadka ₹280/person 4.1

Garbar Burgers ₹200/person 4.6

Baco Tell ₹300/person 3.4

Heera Mahal ₹300/person 4.2

Smokin' Chik ₹250/person 4.0

Online Grocery Delivery Android App

Description:

Developed a grocery delivery app using Jetpack Compose and Kotlin. Users can browse through different categories of groceries, add items to their cart, and place orders. The app integrates Firebase for storage and authentication and follows modern Android development practices.

Key Technologies:
Kotlin, Jetpack Compose, Firebase, MVVM Architecture, Coroutines, LiveData & Flow, Room, Retrofit, Persistence Store

Key Features:

- Browse grocery categories
- Search functionality
- Add items to cart
- Place orders
- User authentication

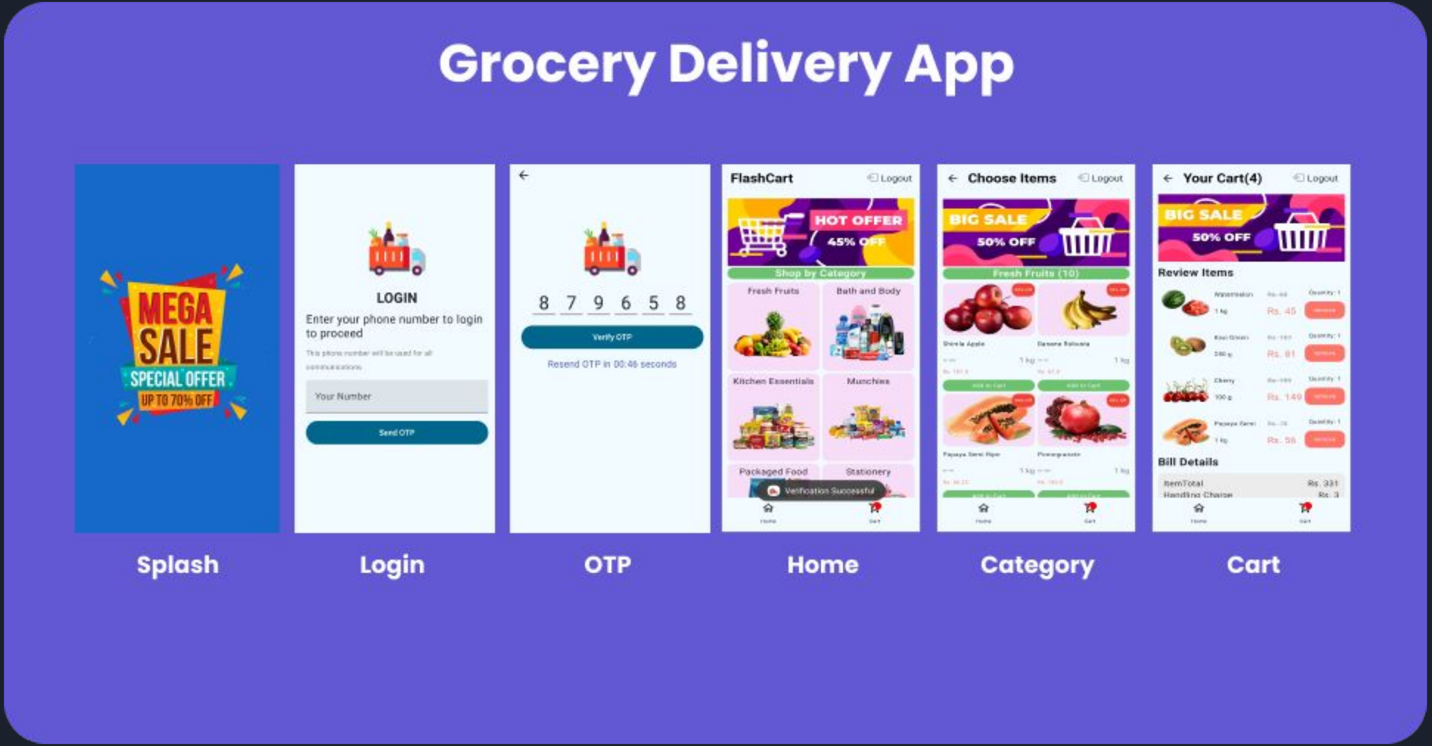
Online Grocery Delivery Android App

Technical Implementations:

- Firebase Integration: Utilized Firebase for user authentication and data storage, providing a secure and reliable backend.
- MVVM Architecture: Implemented the MVVM architecture to ensure a clear separation of concerns and ease of testing.
- Jetpack Compose: Leveraged Jetpack Compose for building responsive and modern UI components.
- Coroutines and LiveData: Used Coroutines for managing background threads and LiveData for reactive UI updates.
- Local Database Management: Integrated Room for local data storage, enabling offline functionality and data persistence.
- Network Requests: Used Retrofit for handling API requests and communicating with the backend services.

Link: Github (https://github.com/arshakshan/Online-Grocery-Shopping-Android-App)

Online Grocery Delivery Android App



LOGIN

Enter your phone number to login to proceed

This phone number will be used for all communications

Your Number

Send OTP

8 7 9 6 5 8

Verify OTP

Resend OTP in 00:46 seconds

FlashCart Logout

HOT OFFER 45% OFF

Shop by Category

Fresh Fruits | Bath and Body | Kitchen Essentials | Munchies | Packaged Food | Stationery

Verification Successful

Choose Items Logout

BIG SALE 50% OFF

Fresh Fruits (10)

Your Cart(4) Logout

BIG SALE 50% OFF

Review Items

Rs. 45

Rs. 81

Rs. 149

Rs. 56

Bill Details

ItemTotal Rs. 331

Handling Charge Rs. 3

MetaVersus - Front End Web App

Description:

Metaverse is a sophisticated web application developed using Next.js and React, designed to provide a seamless, interactive, and responsive user experience. This project emphasizes clean code practices, robust architecture, and modern web development techniques, making it an excellent example of a well-engineered web application.

Key Technologies:
Next.js, React, Tailwind CSS, Framer Motion, Vercel.

Key Features:

- User-friendly interface
- Seamless animations and transitions
- Efficient project structure and code practices

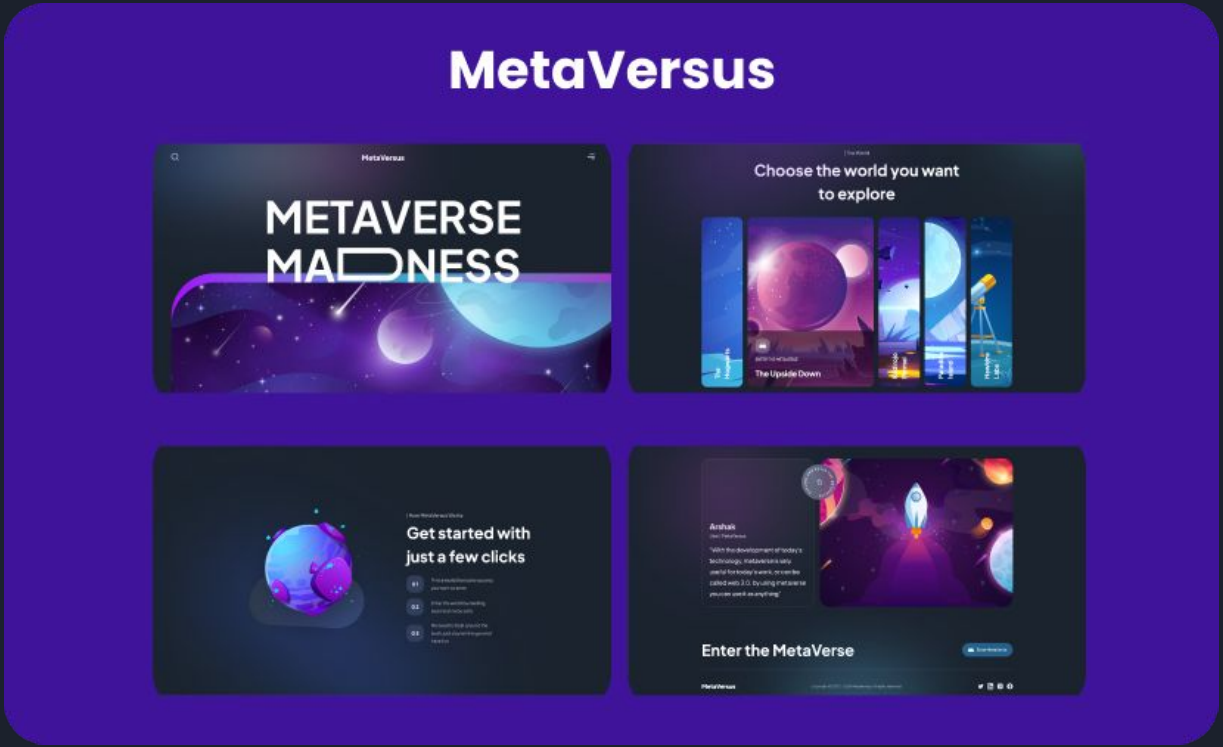
MetaVersus - Front End Web App

Technical Implementations:

- User Interface: Created a user-friendly interface using Next.js and React, ensuring a smooth and intuitive user experience with fast page loads and seamless navigation.
- Styling and Responsiveness: Implemented Tailwind CSS for consistent styling and responsiveness across devices, utilizing media queries for optimal performance on various screen sizes.
- Animations: Integrated Framer Motion to add sophisticated animations and transitions, enhancing the overall user experience with smooth and visually appealing effects.
- Project Structure: Adopted best practices in project architecture, organizing code in a clean and maintainable manner, facilitating easier updates and scalability.
- Development Workflow: Utilized modern development workflows, including linting with ESLint, and efficient package management with npm/yarn, ensuring code quality and consistency.

Links: Github (https://github.com/arshakshan/MetaVersus-Front-End-Web-App)

MetaVersus - Front End Web App



MetaVersus

METAVERSE MADNESS

Choose the world you want to explore

The Upside Down

Get started with just a few clicks

01

02

03

Arshak

"With the development of today's technology, metaverse is very useful for today's work, or can be called web 3.0. by using metaverse you can use it as anything"

Enter the MetaVerse

MetaVersus

Promptopia - Next.js Full Stack App

Description:

Developed a social platform for sharing and discovering creative prompts with authentication, database storage, and CRUD functionalities. The platform allows users to create, edit, and delete prompts, and view prompts from all users.

Key Technologies:
Next.js, React, MongoDB, NextAuth

Key Features:

- Google authentication via NextAuth
- MongoDB database for storing prompts
- Retrieval of prompts for all users and individual users
- Editing and deleting prompts by the creators

Promptopia - Next.js Full Stack App

Technical Implementations:

- Authentication: Integrated Google authentication using NextAuth, allowing secure login and seamless user experience.
- Database: Configured MongoDB to store user prompts, ensuring scalability and efficient data retrieval.
- CRUD Operations: Implemented functionalities to create, read, update, and delete prompts, enabling users to manage their content effortlessly.
- User Profiles: Designed user profiles to display all prompts created by the user, enhancing user interaction and engagement.

Things to Implement:

- Search Functionality:
  - Implement search by prompt, user, and tag to improve discoverability.
- Tag Click:
  - Enable clicking on tags to view all associated prompts, enhancing navigation.
- View Other User Profiles:
  - Develop functionality to view profiles of other users, showcasing their prompts and information.

Link: GitHub (https://github.com/arshakshan/Prompt-sharing-platform)

Promptopia - Next.js Full Stack App

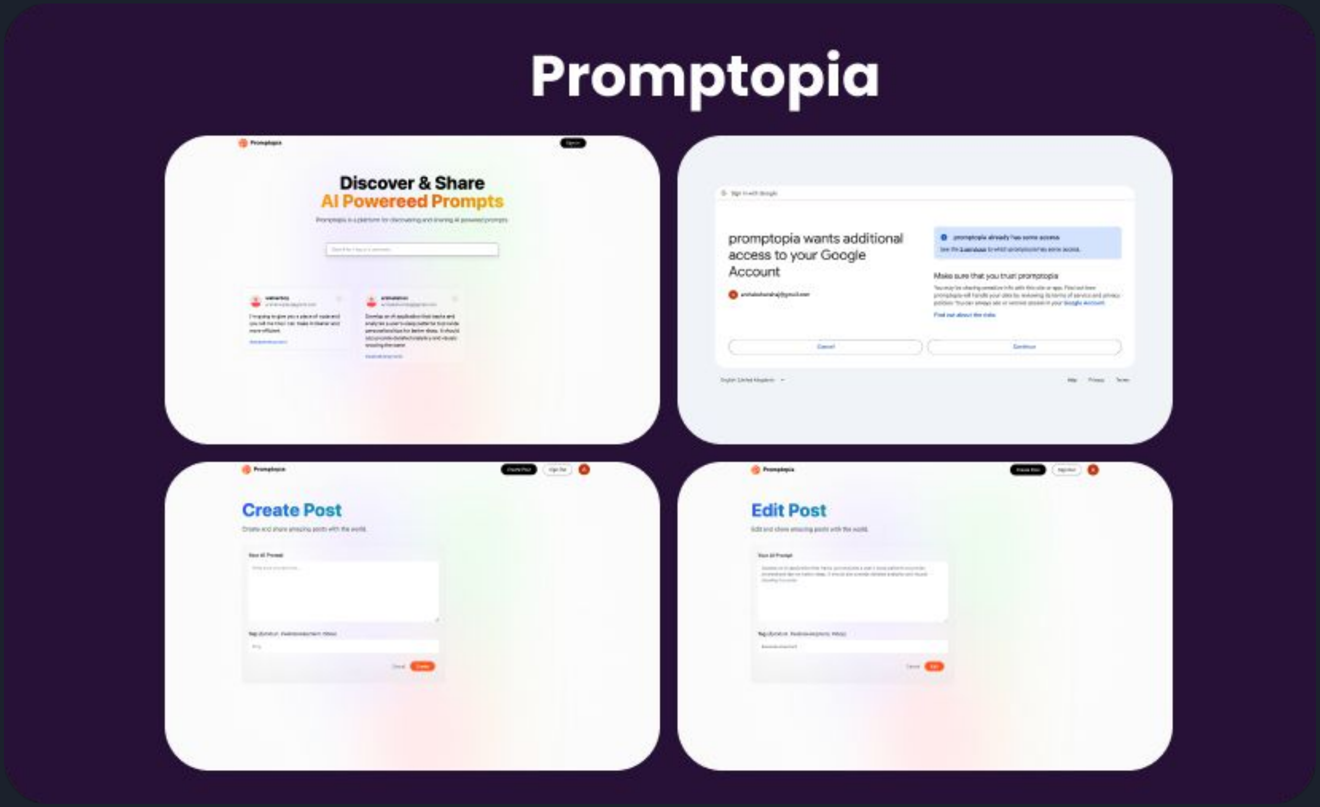
To Run Add the following to the .env file:

```
GOOGLE_ID=356905107284-vilra28dcov0suqdoecoejvbfqknklrk.apps.googleusercontent.com
GOOGLE_CLIENT_SECRET=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

MONGODB_URI=mongodb+srv://arshak:Cw5xsqLkazziRCWq@promptopia.eru9gq5.mongodb.net/?retry
Writes=true&w=majority&appName=promptopia

NEXTAUTH_URL=http://localhost:3000
NEXTAUTH_URL_INTERNAL=http://localhost:3000
NEXTAUTH_SECRET=ec1dR7BlthuATFLeShqEX7hSwAiyfxTijSGdsDb3Dzw=
```

Promptopia - Next.js Full Stack App


Promptopia
Discover & Share
AI Powereed Prompts
promptopia wants additional access to your Google Account
Make sure that you trust promptopia
Cancel
Continue
Create Post
Edit Post

Hotel Reservation Website: (Web Development)

Description:

Developed an intuitive online hotel reservation platform with interactive user interfaces, seamless booking processes, and reliable data management.

Key Technologies:
JavaScript, jQuery, PHP, HTML, MySQL, CSS.

Key Features:

- Interactive user interface
- Seamless booking process
- Reliable data management with MySQL
- Styling with CSS
- Functionality with JavaScript and PHP

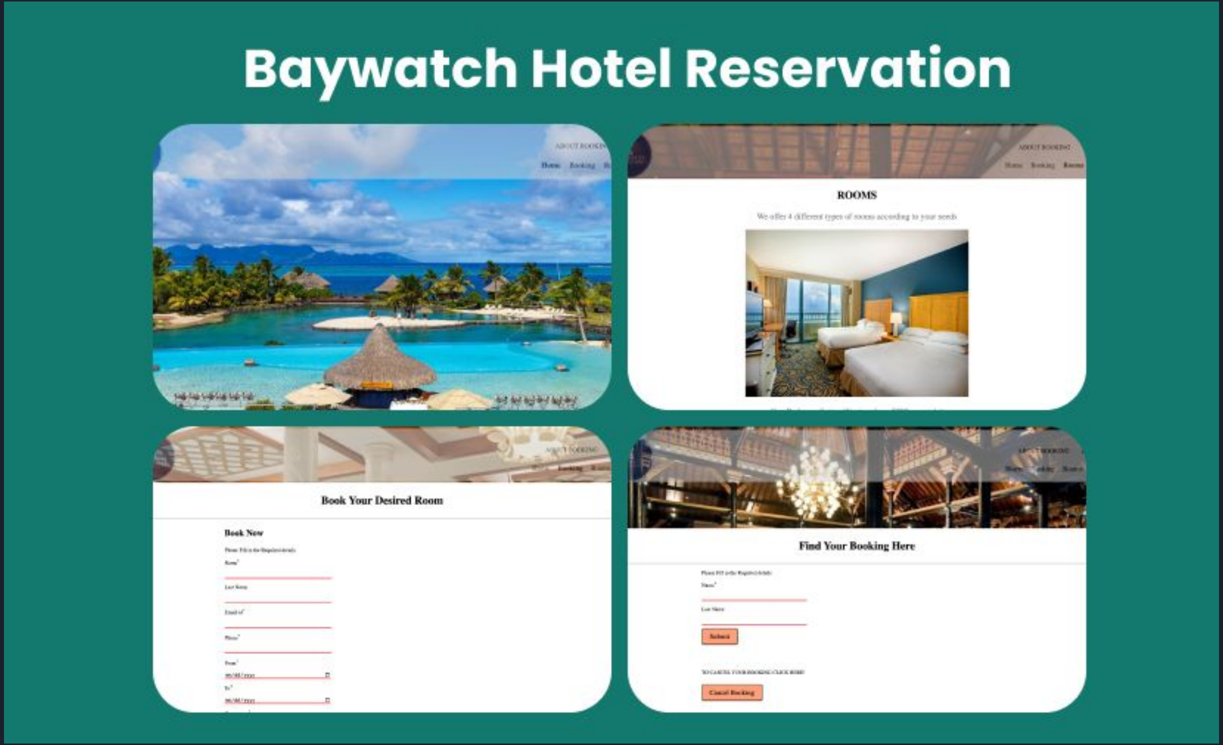
Hotel Reservation Website: (Web Development)

Technical Implementations:

- Interactive User Interface: Designed an engaging and visually appealing user interface using HTML, CSS, and JavaScript, simplifying the reservation process.
- Backend Development: Utilized PHP for server-side scripting, ensuring the platform is dynamic and responsive to user inputs and requests.
- Seamless Booking: Developed a straightforward room selection and reservation system, reducing the complexity often associated with hotel bookings.
- Data Management: Integrated with a MySQL database, guaranteeing reliable storage, retrieval, and management of reservation records.

Links: Github (https://github.com/arshakshan/Baywatch-Hotel-Reservation)

Hotel Reservation Website: (Web Development)



Baywatch Hotel Reservation

ROOMS

We offer 4 different types of rooms according to your needs

Book Your Desired Room

Book Now

Please Fill in the Required details

Name*

Last Name

Email id*

Phone*

From*

mm/dd/yyyy

To*

mm/dd/yyyy

Find Your Booking Here

Please Fill in the Required details

Name*

Last Name

TO CANCEL YOUR BOOKING CLICK HERE!

Deep fake Detection Model (Machine Learning)

Description:

Implemented a dual-stream ML architecture using SRM and RGB streams to distinguish deepfake from genuine videos with 92% accuracy. Used interpretable deep learning for insight-driven analysis.

Key Technologies:
Python, Deep Learning, Machine Learning, Jupyter Notebooks.

Key Features:

- Dual-stream architecture
- Interpretable deep learning
- Computer Vision

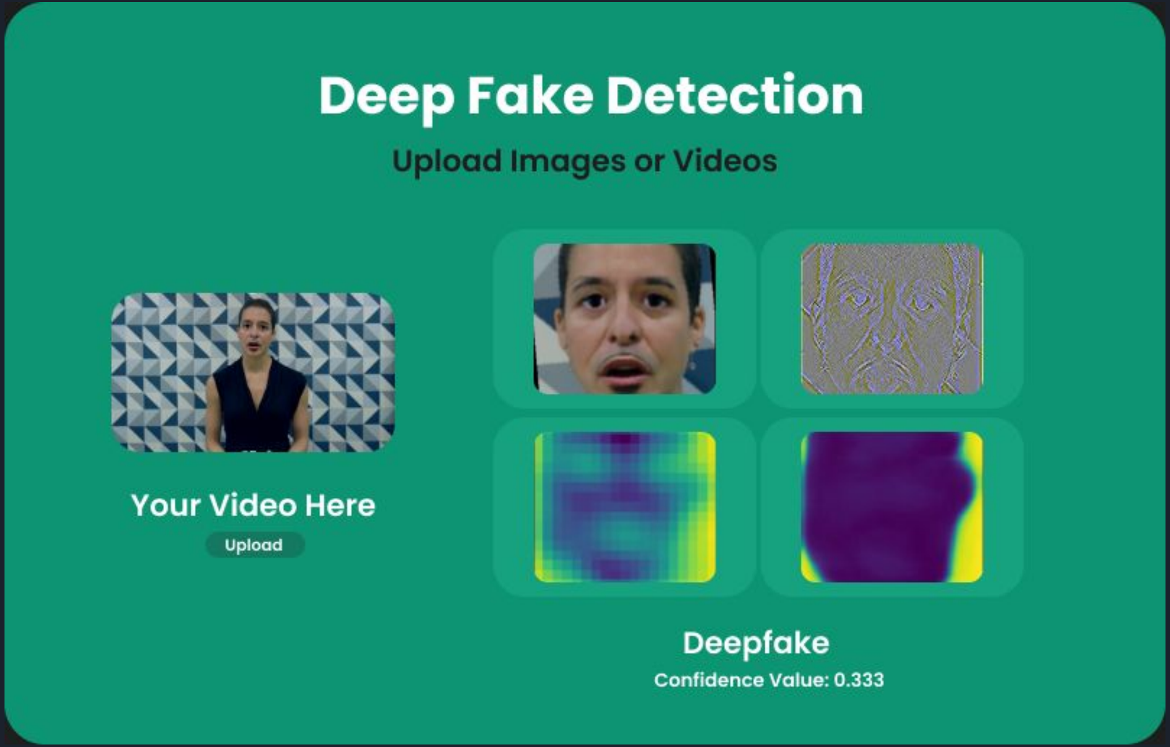
Deep fake Detection Model (Machine Learning)

Technical Implementations:

- Dual-Stream Architecture: Combined Spatial Rich Model (SRM) and RGB streams to distinguish deep fake from authentic videos.
- Accuracy: Achieved a 92% accuracy rate in distinguishing deepfakes from genuine videos.
- Interpretable Deep Learning: Utilized interpretable deep learning to provide insight-driven analysis, highlighting key features in videos that informed model decisions.
- Python Libraries: Employed advanced Python libraries to reinforce the model's applicability and efficiency.
- Documentation: Authored comprehensive documentation detailing project methodologies and outcomes, ensuring clarity and reproducibility.

Links: Github (https://github.com/arshakshan/Deepfake-Detection)

Deep fake Detection Model (Machine Learning)



Deep Fake Detection

Upload Images or Videos

Your Video Here

Deepfake

Confidence Value: 0.333

COVID-19 Case Prediction Model (Machine Learning)

Description:

Designed a COVID-19 case predictor using the John Hopkins Database. Incorporated dual regression techniques for accurate forecasts.

Key Technologies:
Python, Pandas, Machine Learning, Time Series Analysis.

Key Features:

- Dual regression techniques
- 20-day forecast capability
- John Hopkins Live Data Point
- Exploratory Data Analysis
- Use of various Machine Learning Models like Polynomial Regression, Support Vector Regression

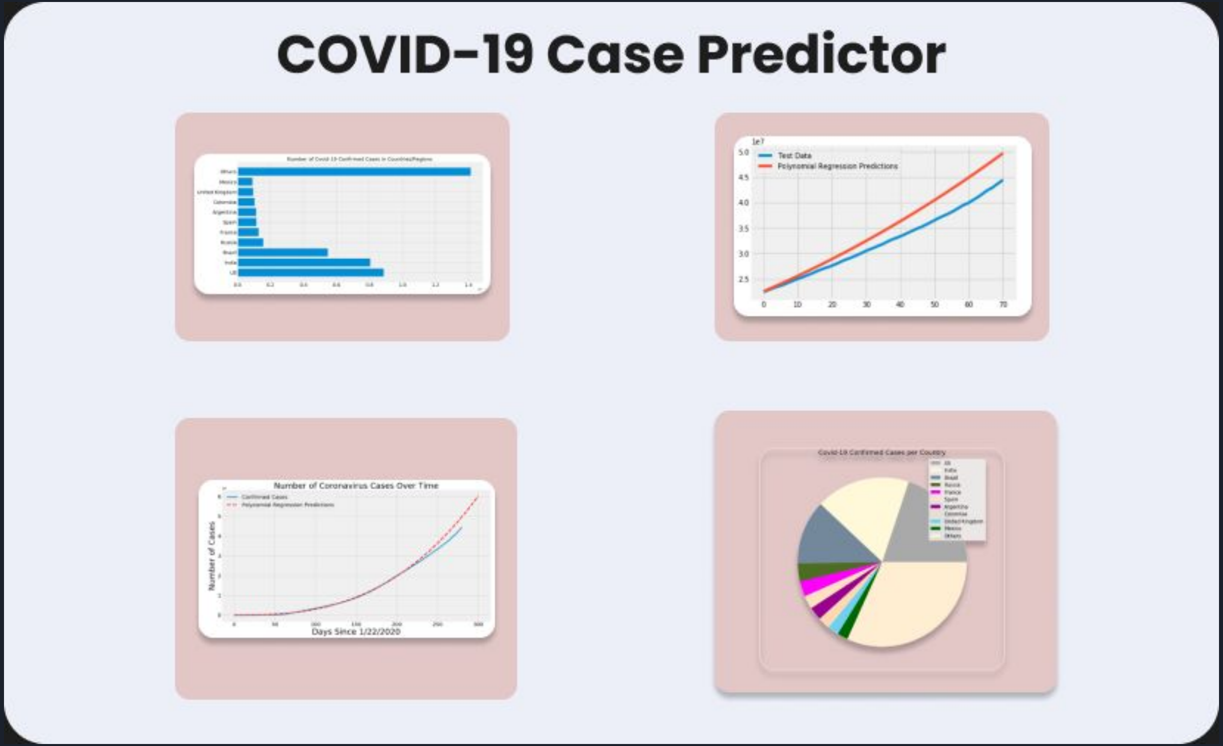
COVID-19 Case Prediction Model (Machine Learning)

Technical Implementations:

- Dual Regression Techniques: Incorporated Polynomial Regression (PolyReg) and Support Vector Regression (SVR) models, achieving outstanding prediction accuracy rates of 96.44% and 97.38%, respectively.
- Exploratory Data Analysis (EDA): Performed in-depth EDA on data from various countries using matplotlib and seaborn, ensuring comprehensive insights into regional trends and patterns.
- 20-Day Forecast: Enabled short-term forecasts for up to 20 days, providing policymakers, healthcare professionals, and the general public with invaluable outlooks during the pandemic.
- Data Visualization: Utilized Python for data visualization, presenting insights and trends clearly and effectively.

Links: Github(https://github.com/arshakshan/covid-19-case-prediction)

COVID-19 Case Prediction Model (Machine Learning)



Lung Cancer Segmentation & Classification (Machine Learning)

Description:

Conducted research on lung cancer segmentation and classification, achieving a 97.25% accuracy using the Cuckoo Search Algorithm on the ELCAP dataset. Published findings in the International Journal of Enhanced Research in Science, Technology & Engineering.

Key Technologies:
Cuckoo Search Algorithm, MATLAB, Swarm Intelligence.

Key Features:

- High accuracy in segmentation and classification
- Advanced algorithm application
- MATLAB implementation

Lung Cancer Segmentation & Classification (Machine Learning)

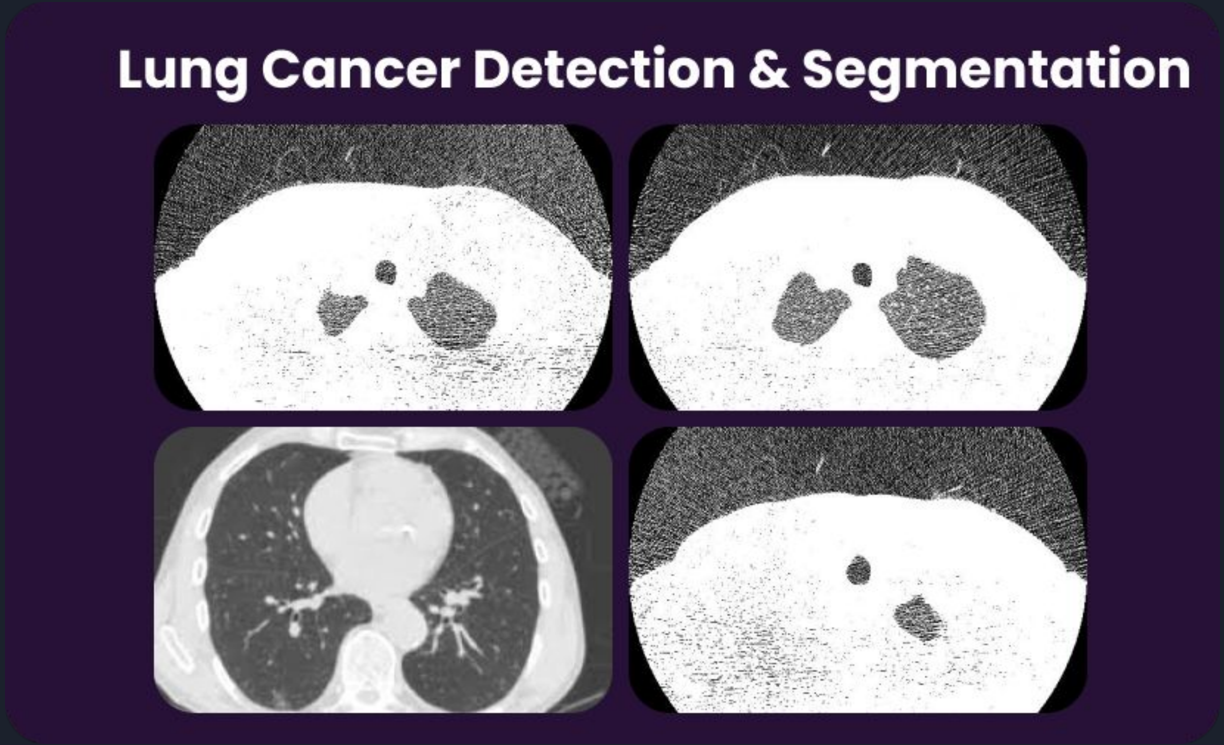
Technical Implementations:

- Cuckoo Search Algorithm: Applied the Cuckoo Search Algorithm for lung cancer segmentation, achieving a high accuracy rate of 97.25%.
- ELCAP Dataset: Utilized the Early Lung Cancer Action Program (ELCAP) dataset, which provided a comprehensive set of images for training and testing the algorithm.
- Research Publication: Published the research findings in the International Journal of Enhanced Research in Science, Technology & Engineering, highlighting the effectiveness of the Cuckoo Search Algorithm in medical imaging.
- MATLAB Implementation: Implemented the algorithm in MATLAB, ensuring reproducibility and scalability of the solution.

Links:

- Github (https://github.com/arshakshan/Lung-Cancer-Segmentation)
- Publication (Link)

Lung Cancer Segmentation & Classification (Machine Learning)



Lights Out: Forgotten Room (Unity Game)

Description:

Spearheaded the development of a Unity-based survival horror game, implementing complex light mechanics, dynamic difficulty systems, and immersive sound design. Conducted extensive testing for smooth performance across devices.

Key Technologies:
Unity, C#, Unity Engine.

Key Features:

- Complex light mechanics
- Dynamic difficulty system
- Atmospheric sound design
- Interactive game objects and puzzles

Link: GitHub (https://github.com/arshakshan/Lights-Out-Game)

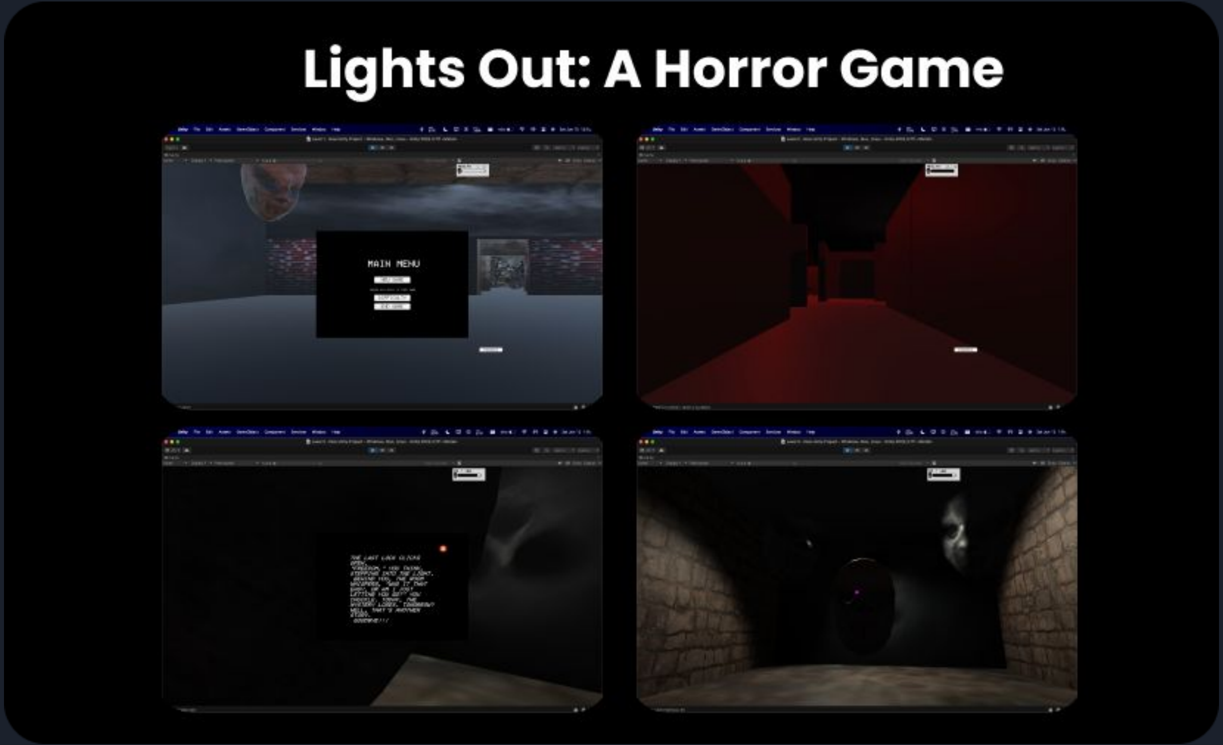
Lights Out: Forgotten Room (Unity Game)

Technical Implementations:

- Light Mechanics: Engineered complex light mechanics, including torch usage and ambient flickering effects, to reveal hidden clues and create suspenseful gameplay.
- Dynamic Difficulty: Implemented a dynamic difficulty system that adapts the game's challenges based on the player's performance, providing a tailored experience.
- Sound Design: Crafted atmospheric elements using advanced sound design techniques, integrating ambient sounds and jump scares that dynamically respond to player actions and environmental changes.
- Interactive Objects and Puzzles: Designed and coded interactive game objects and environmental puzzles requiring logical thinking and physical interaction, contributing to diverse gameplay and replay value.
- Collaboration: Worked with a team of artists and other developers to ensure cohesive and visually striking horror aesthetic.

Link Link: GitHub (https://github.com/arshakshan/Lights-Out-Game)

Lights Out: Forgotten Room (Unity Game)



Lights Out: A Horror Game

THE LAST LOCK CLICKS OPEN. "FREEDOM," YOU THINK, STEPPING INTO THE LIGHT. BEHIND YOU, THE VOICE WHISPERS, "WAS IT THAT EASY, OR AM I JUST LETTING YOU GO?" YOU [illegible]. TODAY, THE MYSTERY LIVES. TOMORROW? WELL, THAT'S ANOTHER STORY. GOODBYE!!!