

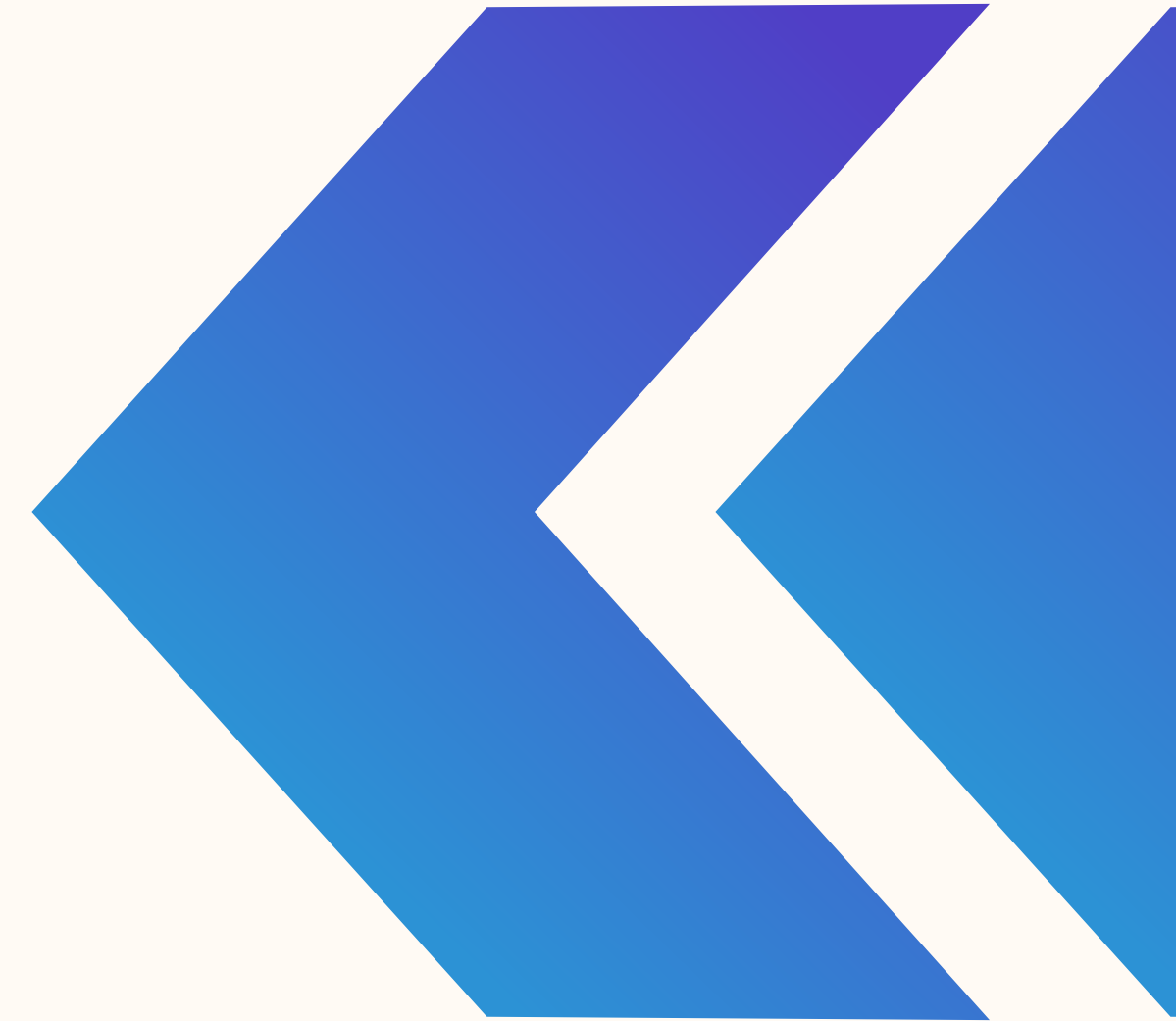


Team Maverick's AI-driven Solution

PRESENTED BY:

Bhaskar Pranav

Shardul Varin



Problem Insight & Proposed System Architecture

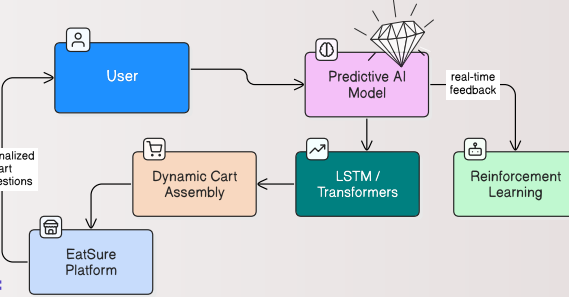
AIM

Reduce cart processing time (CPT) (from app login to checkout) to improve efficiency and customer satisfaction.

CART PROCESSING TIME ANALYSIS

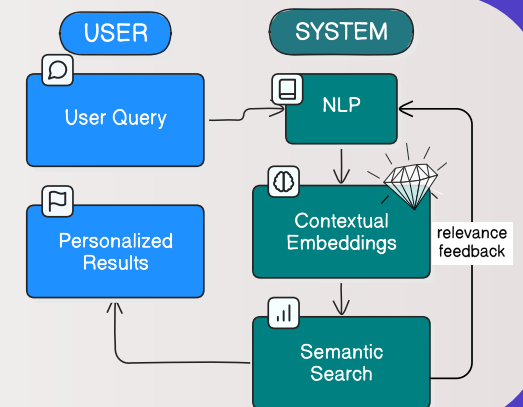
Issue : Item Selection for Cart

Solution: AI-driven Predictive Cart Assembly system to leverage time series forecasting, deep learning (LSTM, Transformers), and reinforcement learning to predict and pre-fill carts based on user history, time of day, and behavioral patterns.



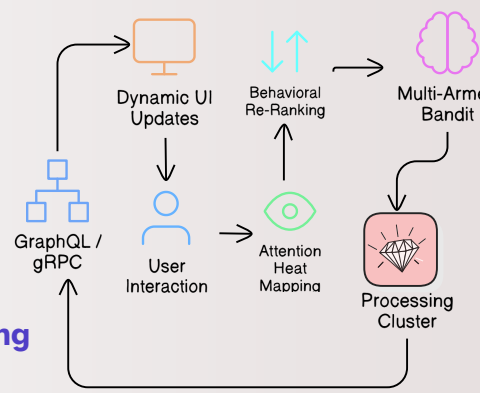
Issue : Lack of Personalization

Solution: Hyper Personalized Neural Search (by Custom LLM trained on 50M+ food items to answer conversational queries like "What should I eat after workout")



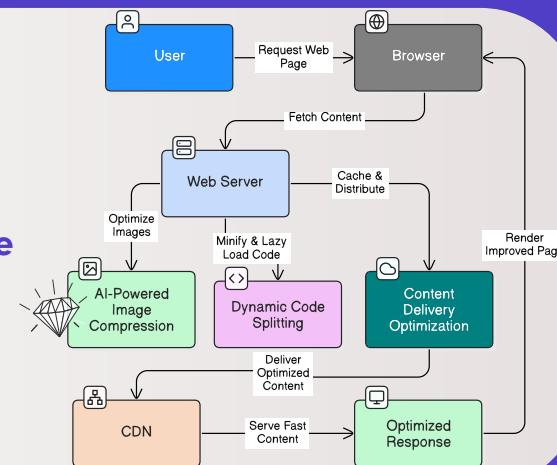
Issue : Static UI Optimization

Solution : A server-driven UI adapts in real-time using attention heat mapping, optimizing layout via behavioral re-ranking and Multi-Armed Bandit models, with GraphQL/gRPC delivering dynamic UI updates.



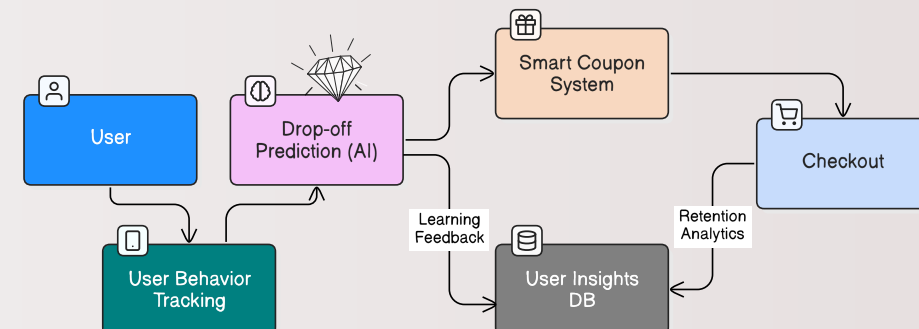
Issue : Bad Website Performance

Solution: AI-powered image compression, code splitting, and CDN optimization for improved performance



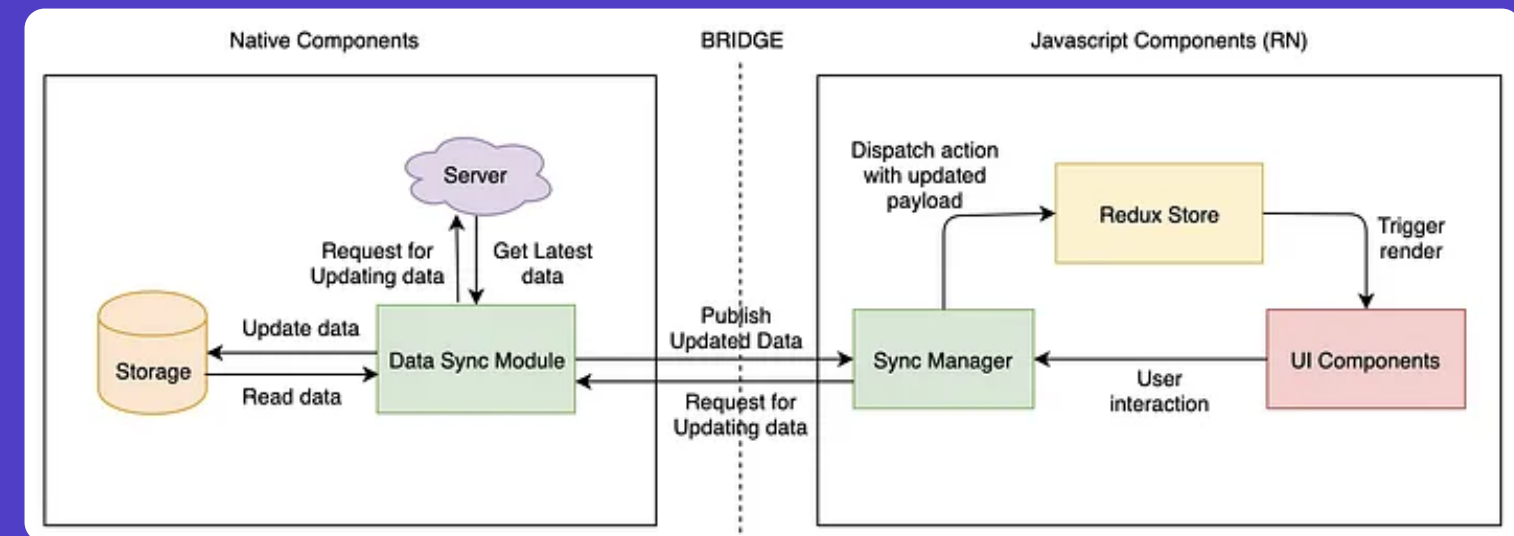
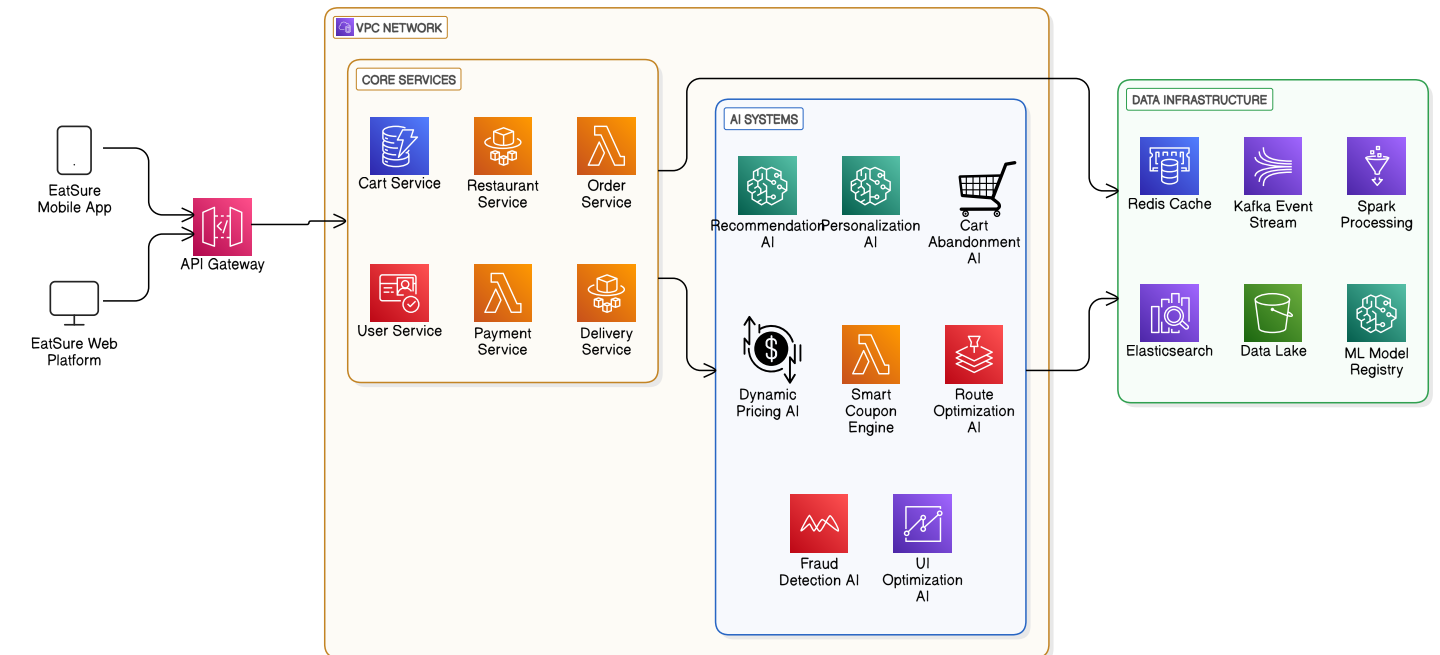
Issue : Cart Dropping By User

Solution: An AI-based model which tracks user journey from opening the app to placing the order. The model shall recognize the phase where the user is most likely to leave the app and shall suggest coupons at that stage to keep the user hooked.



JS Native Components: Handle core functionalities like data synchronization, analytics, fraud detection, and system utilities, structured into multiple layers.

- JavaScript Components: Manage UI interactions, state synchronization via Redux, and logging/monitoring, interfacing with native modules through the React Native Bridge.



Component Layers:

- Divided into multiple layers: Business Intelligence, Application Core, Frameworks/Modules, and Commons.
- Ensures modular development and separation of concerns.

React Native Bridge Integration:

- Connects UI Components with the Backend Data Sync and Redux Store.
- Facilitates interaction between native modules and JavaScript components for real-time updates.

Development Plan, Resource Estimates & Success Metrics



Development Plan

Research and Design

Focus: Cart Abandonment Rate (-68%)

- User journey mapping and bottleneck identification
- AI model selection and prototype development
- Establish baseline measurements for all KPIs

1

Duration
1 month

Deliverable:
Comprehensive user journey map
with identified friction points

End of Phase 1

MVP Implementation

Focus: Cart Processing Time (-65%)

- Core recommendation engine development
- Personalization
- Algorithm implementation
- A/B-testing with 5% user base
- Optimize Checkout Algo

2

Duration
2 months

Deliverable:
Functional MVP with 20-30%
improvement in processing time

End of Phase 2

Full Deployment

Focus:
Order Completion Rate (+40%)
User Engagement (+55%)
Recommendation Accuracy (+42%)

- Full-platform implementation and integration
- System optimization and API development
- A/B testing and iterative improvements
- Performance monitoring and adjustments

3

Duration
4 months

Deliverable:
Fully deployed system meeting
all KPI targets

Completion



Resource Estimates

Financial Investment

- Artificial Intelligence & Machine Learning Infrastructure: ₹1.5-2 Crore
 - Cloud Computing Resources: ₹80 Lakhs-1 Crore
 - Development Team Costs: ₹1.06-1.66 Crore
 - Other Development Costs: ₹40-60 Lakhs
 - Contingency Budget: ₹30-50 Lakhs
- Total Project Investment: ₹5.76 Crore



Manpower Estimates

AI Architects – Neural network specialists who will craft our recommendation algorithms and personalization engines

Integration Engineer – The vital bridge between AI models and EatSure's existing systems

Experience Developer – Translates AI capabilities into intuitive front-end interactions

Cloud Infrastructure Strategist – Designs scalable architecture that grows with user demand



AI tools



HIGH-LEVEL API FOR TENSORFLOW TO
SPEED UP NEURAL NETWORK
DEVELOPMENT



DEEP LEARNING FRAMEWORKS
FOR BUILDING AND OPTIMIZING
RECOMMENDATION ALGORITHMS



VERSIONING AI DATASETS



MODEL TRACKING AND
DEPLOYMENT AUTOMATION



Success Metrics



Projected improvements after implementation of Adaptive Neural Cart System (ANCS)



Expected Results



Cart Processing Time (-65%):
Time from app open to checkout
completion



Cart Abandonment Rate (<15%):
Users who abandon items in cart



Order Completion Rate (+40%):
Successfully completed
transactions



User Engagement (+55%):
Weekly active user interactions



Recommendation Accuracy (92%):
Personalized suggestion relevance



Detailed Development Plan

1

Research and Design

Focus:
Cart Abandonment Rate
Cart Processing Time
Identify further problems

User journey mapping and bottleneck identification

- Use NLP to check if there are any particular complaints via app store reviews /surveys / other modes
- Segment journey into phases and check which phase has highest drop off rate

Establish baseline measurements for all KPIs

- Cart Processing Time - self explanatory
- Cart Abandonment - Step-by-Step Abandonment Rate
- User Engagement - Number of Returns
- Recommendation Accuracy - Revenue from such items

Deliverable:
Comprehensive user journey map with identified friction points

End of Phase 1

Duration 1 month

2

MVP Implementation

Focus:
Help and Support
Slow Delivery

Help and Support

- Auto Resolutions for low-risk cases using business rules.
- Real-Time Context Awareness from order data and delivery status.
- Evidence Handling via image uploads and metadata checks.
- Smart Escalations triggered by sentiment and case complexity.
- Feedback-Driven Learning to continuously refine support logic.

Slow Delivery

- 3 key factors - Resto proximity , Rider assigned , and Prep Time.
- Use XGBoost/ Random forest to check whether order will be on time or likely to be delayed.
- Input data to model: Historical multi-resto orders, assigned delivery time, actual delivery time
- Include rush factor

Deliverable:
Comprehensive user journey map with identified friction points

End of Phase 2.1

Duration 1 month

Research and Design

Focus:
Cart Abandonment Rate
Cart Processing Time
Help and Support
Slow Delivery

User Engagement

- Personalized food recommendations (Swiggy Sixes, Zomato Flash Sale)
- Focus on number of returns
- Classify restaurants based on reviews and flag them for improvement.
- If order cancelled due to delivery issues, then focus on the "Slow Delivery" point

Recommendation Accuracy

- Use Ranking Model for relevant items [P(add) + profit margin]
- Offer micro-discounts
- A/B Testing + ML to learn best layout and screen

Deliverable:
Comprehensive user journey map with identified friction points

End of Phase 2.2

Duration 1 month



Detailed Development Plan

Foundation

- Deploy containerized microservices architecture
- Integrate ranking algorithm with product database
- Set up user behavior tracking system
- Implement basic chatbot functionality
- Configure delivery zone management

Personalization

- Configure existing recommendation models with reinforcement learning
- Implement user-specific personalization
- Optimize API response times
- Deploy existing context-aware help system
- Integrate existing coupon suggestion system

3

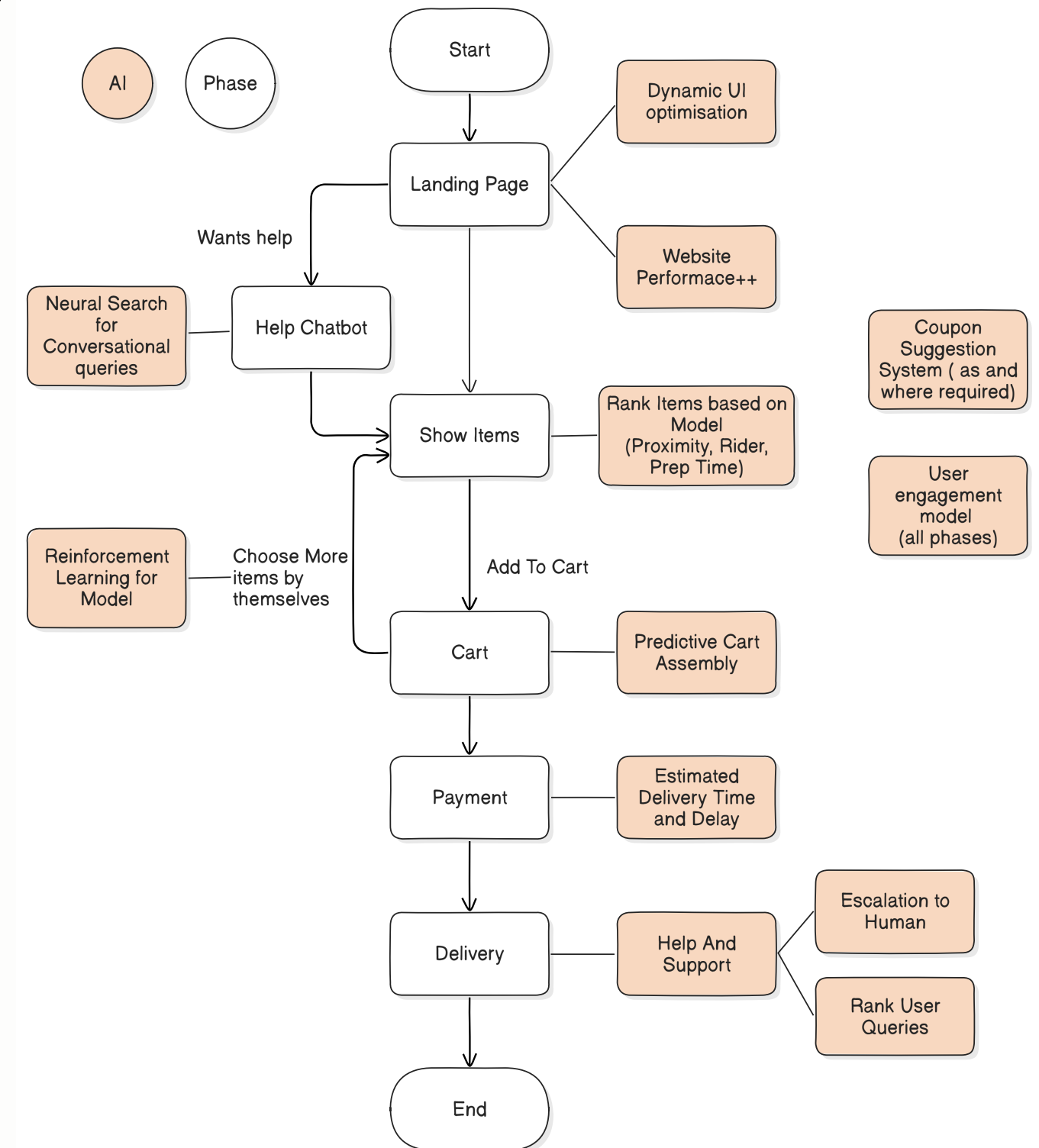
Intelligence Layer

- Deploy existing ML models for item ranking
- Implement existing predictive cart assembly
- Establish A/B testing infrastructure
- Integrate pre-built delivery prediction system
- Create performance monitoring dashboards

Scale And Refinement

- Fine-tune deployed ML models with production data
- Implement automated anomaly detection
- Complete performance optimization
- Deploy end-to-end engagement tracking
- Establish continuous improvement framework

High Level View





Potential Challenges & Mitigation Strategies



Issue: AI Accuracy & Performance

Challenges:

- Building AI models that provide truly relevant recommendations
- Cold start problem for new users with limited history
- Maintaining real-time performance at scale

Mitigation Strategies:

- Use hybrid recommendation systems (collaborative + content-based)
- Implement A/B testing frameworks to continuously optimize models
- Develop fallback mechanisms for new users (category-based recommendations)
- Set up model monitoring dashboards to detect performance degradation
- Establish regular model retraining pipelines



Issue: Technical Integration with current capabilities

Challenges:

- Integrating AI systems with existing legacy infrastructure
- Ensuring smooth data flow between components
- Managing backend-frontend communication efficiently

Mitigation Strategies:

- Create well-defined APIs for system integration
- Implement event-driven architecture for real-time updates
- Develop comprehensive documentation for all integration points
- Use feature flags to gradually roll out new capabilities
- Establish a robust CI/CD pipeline for seamless deployment



Issue: Data Privacy & Security

Challenges:

- Collecting and processing user behavior data raises privacy concerns
- Compliance with data protection regulations (like GDPR)
- Securing sensitive payment information

Mitigation Strategies:

- Implement data anonymization techniques
- Deploy end-to-end encryption for all user data
- Establish clear consent mechanisms for data collection
- Conduct regular security audits and penetration testing
- Develop a comprehensive privacy policy



Issue: Cost Management and Allocation

Challenges:

- Balancing development costs with expected ROI
- Managing ongoing cloud infrastructure expenses
- Controlling costs for AI/ML model training and inference

Mitigation Strategies:

- Implement cost monitoring and optimization tools
- Use serverless computing where appropriate to reduce idle resources
- Establish clear budget thresholds and alerts
- Balance model complexity with operational costs
- Prioritize features based on ROI potential



Issue: Scalability & Performance

Challenges:

- Handling peak traffic during high-demand periods
- Minimizing latency for real-time recommendations
- Ensuring system stability under increasing user load

Mitigation Strategies:

- Design cloud-native architecture with auto-scaling capabilities
- Implement caching mechanisms for frequently accessed data
- Use CDN for content delivery optimization
- Adopt microservices architecture for independent scaling
- Establish load testing as part of the development pipeline



Issue: User Adoption & Experience

Challenges:

- User resistance to AI-powered recommendations
- Balancing automation with user control
- Avoiding UI complexity with new features

Mitigation Strategies:

- Implement gradual rollout with user feedback loops
- Design intuitive UI that makes AI recommendations feel natural
- Provide clear explanations for recommendations
- Allow users to easily modify or override AI suggestions
- Create onboarding experiences highlighting new features

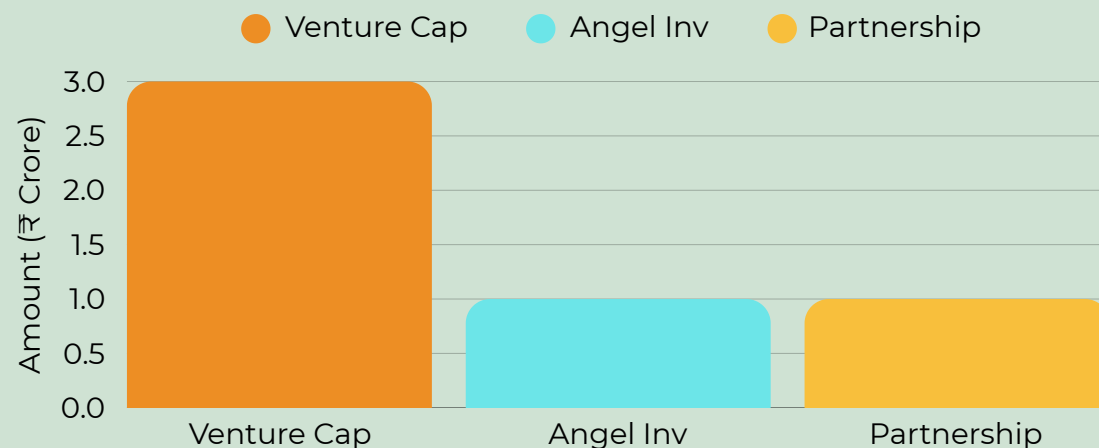


In the pursuit of reducing cart processing time and enhancing user experience, this case study demonstrates how AI can transform operational efficiency at EatSure. By addressing core challenges like non-personalized recommendations, UI friction, and payment delays, and proposing structured mitigation strategies, the solution ensures both scalability and cost-effectiveness.

Funding/Resource Breakdown

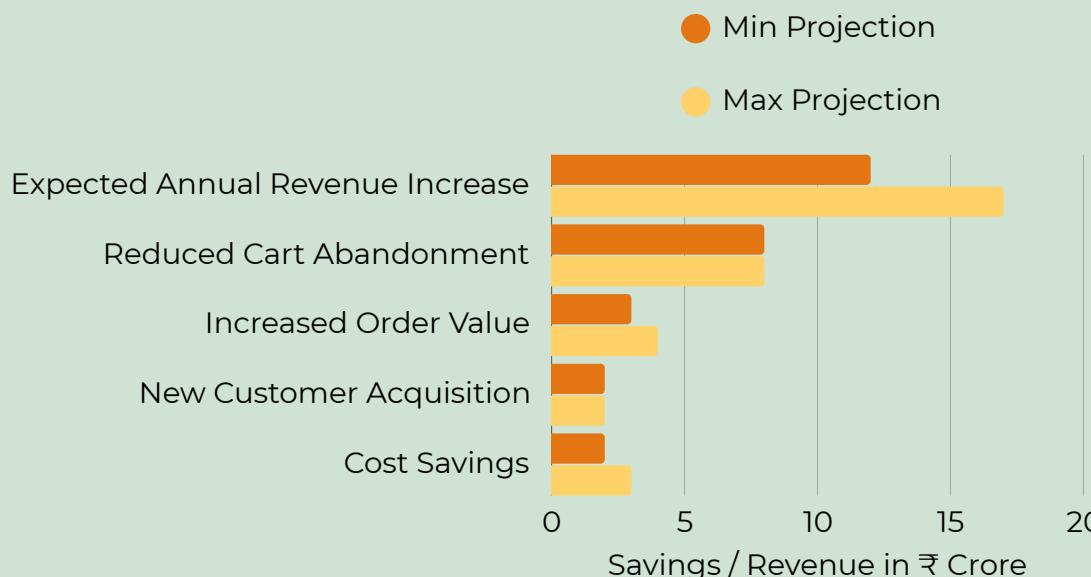
Investment Sources

- **Venture Capital: ₹3.0 Crore (Series A funding)**
 - Focus on AI/ML specialized VCs like Sequoia India, Accel, and Blume Ventures
- **Angel Investors: ₹1.0 Crore (Early-stage capital)**
 - Consortium of 4-5 industry experts with food-tech experience
- **Strategic Partnership: ₹1.0 Crore (Co-development funding)**
 - Technology partnership with major cloud provider (AWS/Azure/GCP)



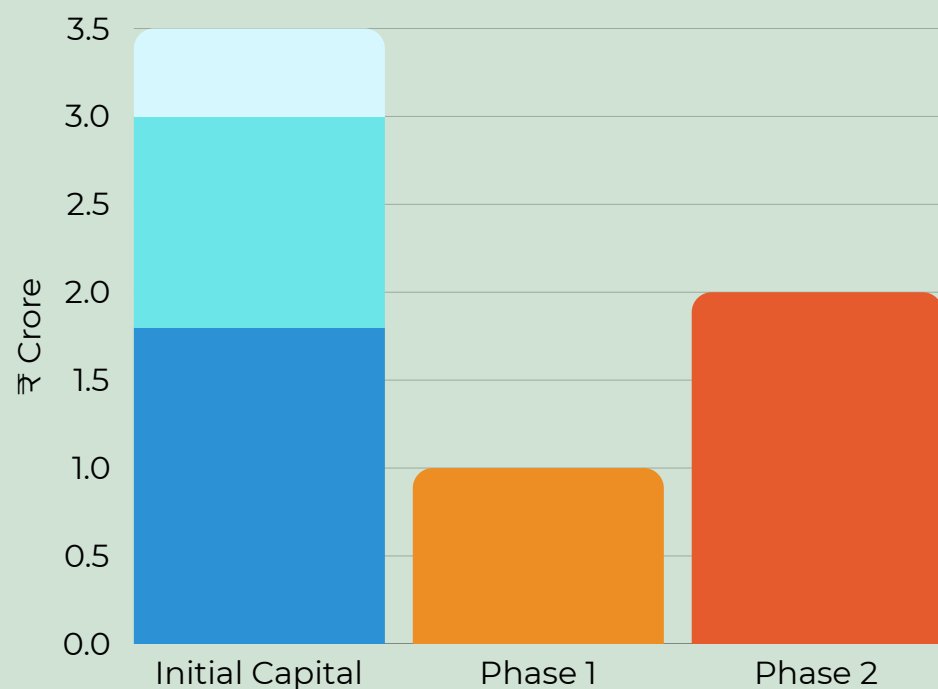
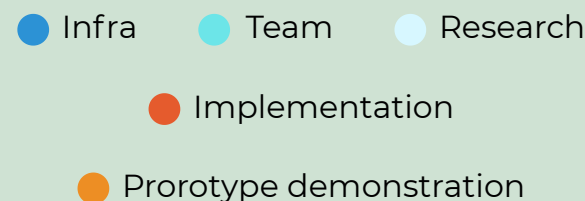
ROI Projections

- **Payback Period: 18-24 months**
- **Projected 3-Year ROI: 200% (₹11.52 Crore return on ₹5.76 Crore investment)**
- **Cost Savings: ₹2-3 Crore annually through operational efficiencies**

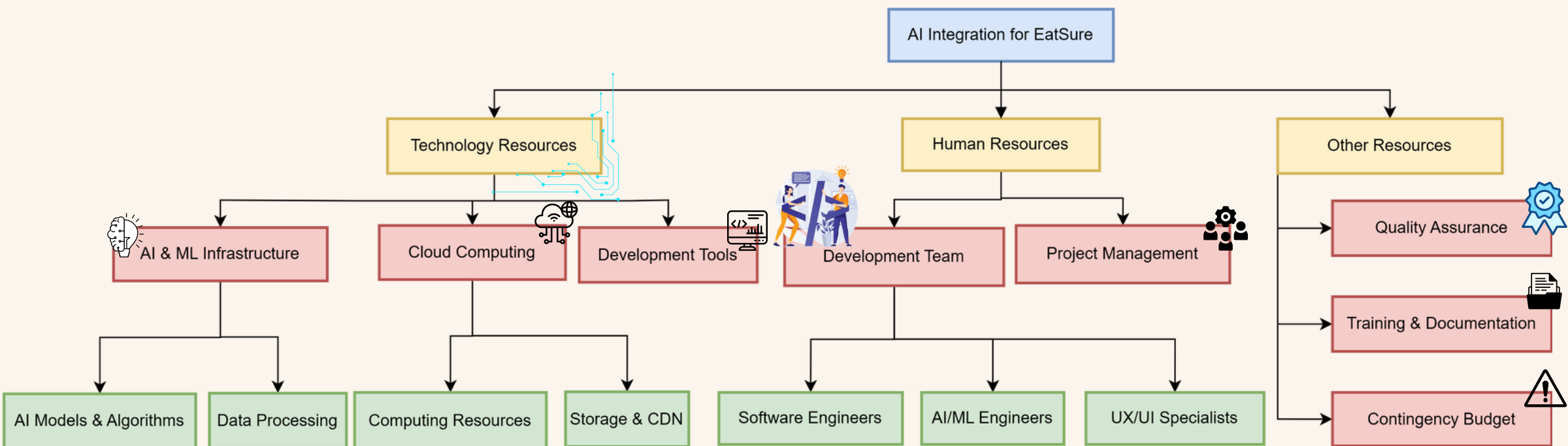


Financial Structure

- **Initial Capital Requirement: ₹3.5 Crore (60% of total)**
 - Development infrastructure: ₹1.8 Crore
 - Team onboarding: ₹1.2 Crore
 - Research & initial prototyping: ₹0.5 Crore
- **Milestone-Based Releases: ₹2.26 Crore (remaining 40%)**
 - **Phase 1 Completion: ₹0.76 Crore (Research and design phase focused on user journey mapping, bottleneck identification, and AI model selection with a functional prototype demonstrating 15% CPT improvement.)**
 - **Phase 2 Completion: ₹1.5 Crore (MVP implementation phase delivering a fully integrated solution with personalized recommendations and dynamic UI optimizations achieving 20-30% improvement in cart processing time.)**



Resource Breakdown Structure



THANK YOU

PRESENTED BY:

Bhaskar Pranav

Shardul Varin

