

# Pickle Rocket - PICKLE - White Paper

therealpicklerocket (AI Generated)

February 2024

## Abstract

Pickle Rocket (PICKLE) is a groundbreaking meme coin that merges the realms of artificial intelligence (AI) and cryptocurrency to pioneer a new era of digital investment and community engagement. As the cryptocurrency market continues to surge, PICKLE capitalizes on the growing excitement surrounding both AI technology and meme coins, providing a platform where users, affectionately referred to as "Lil' Pickies," can actively participate in token distribution, staking, and the creation of AI-generated art projects. The primary objective of PICKLE is twofold: to generate unparalleled hype and interest within the cryptocurrency community, particularly among enthusiasts keen on exploring the intersection of AI and digital assets, and to ride the current bull run in the market by offering a unique and innovative investment opportunity. Through a combination of inventive tokenomics and community-driven initiatives, PICKLE aims to incentivize engagement, foster a strong sense of community, and propel its value to new heights.

## 1 Introduction

The intersection of artificial intelligence (AI) and cryptocurrency represents a fascinating frontier in the ever-evolving landscape of digital innovation [3]. As both AI technology and the cryptocurrency market continue to gain momentum, the potential for synergy and collaboration between these domains is becoming increasingly apparent. It is within this context that Pickle Rocket (PICKLE) emerges as a groundbreaking venture, poised to revolutionize the way we perceive and interact with digital assets.

Cryptocurrency, once a niche interest confined to tech-savvy enthusiasts, has experienced exponential growth in recent years, propelled by mainstream adoption, institutional interest, and a wave of technological innovation [7]. The emergence of meme coins, characterized by their playful branding and community-driven ethos, has further diversified the cryptocurrency landscape, attracting a new wave of investors and enthusiasts eager to participate in the digital gold rush.

Simultaneously, advances in artificial intelligence have captured the imagination of technologists and visionaries worldwide, offering unprecedented opportunities for innovation across various industries. From predictive analytics to natural language processing, AI has demonstrated its transformative potential, reshaping the way we work, communicate, and interact with technology. In the realm of cryptocurrency, AI-powered solutions hold the promise of enhanced security, efficiency, and predictive analysis, providing investors with valuable insights and tools to navigate the volatile market landscape.

PICKLE seeks to harness the power of AI and cryptocurrency to create a dynamic and engaging platform for users, affectionately referred to as "Lil' Pickies," to participate in token distribution,

staking, and the creation of AI-generated art projects. At its core, PICKLE represents a fusion of innovation, community, and speculative fervor, encapsulating the zeitgeist of the contemporary crypto ecosystem.

The primary objective of PICKLE is to generate unparalleled hype and interest within the cryptocurrency community, capitalizing on the growing fascination with both AI technology and meme coins. By offering a novel investment opportunity that combines elements of speculation, community engagement, and technological innovation, PICKLE aims to carve out a unique niche in the competitive cryptocurrency market.

At the heart of PICKLE's ecosystem lies its distinctive features and functionality. The token distribution process, powered by the PICKLE contract, dynamically dispenses tokens to Lil' Pickies based on the prevailing market hype and the inflow of funds. This ensures a responsive and engaging experience for users, as token rewards fluctuate in real-time according to market dynamics.

Moreover, PICKLE introduces a revolutionary staking mechanism whereby Lil' Pickies can stake their PICKLE tokens via the contract, earning bonuses in BNB and additional PICKLE tokens as the protocol grows. A significant portion of the value added to the protocol through staking is reinvested back into the community, serving as a testament to PICKLE's commitment to rewarding long-term engagement and participation. In addition to its innovative tokenomics, PICKLE dedicates a substantial portion of its funds to the creation of AI-generated art projects. These projects, funded through token distribution and staking, are exclusively available for purchase using PICKLE tokens, thereby creating a unique marketplace and driving demand for the currency.

PICKLE further enhances community engagement through a series of airdrops and mini rocket NFT airdrops, providing Lil' Pickies with exclusive opportunities to participate in closed PICKLE auctions and contribute to the growth of the protocol. Also, with its transparent and secure blockchain infrastructure, PICKLE ensures the integrity of all transactions and operations, providing users with confidence in the reliability and security of the platform. By leveraging AI technology to optimize token distribution and create value-added art projects, PICKLE presents a compelling investment opportunity for crypto enthusiasts and investors alike.

In conclusion, Pickle Rocket (PICKLE) represents a pioneering venture into the realms of AI and cryptocurrency, offering a dynamic and engaging platform for users to participate in the crypto revolution. With its innovative features, community-driven initiatives, and commitment to technological advancement, PICKLE is poised to soar to new heights in the ever-evolving landscape of digital assets.

## Purpose and Goals

The purpose and goals of Pickle Rocket (PICKLE) are rooted in the recognition of the transformative potential of both artificial intelligence (AI) and cryptocurrency within the global digital landscape. PICKLE seeks to harness the synergies between these domains to achieve several key objectives, outlined below:

### 1.1 Generate Hype and Interest

PICKLE aims to generate unparalleled hype and interest within the cryptocurrency community, particularly among enthusiasts and investors keen on exploring the intersection of AI technology and digital assets. By leveraging the growing fascination with both AI and meme coins, PICKLE seeks

to capture the imagination of a diverse audience, including tech-savvy investors, crypto enthusiasts, and individuals intrigued by the potential of emerging technologies.

## **1.2 Ride the Bull Run**

In tandem with its goal of generating hype and interest, PICKLE seeks to capitalize on the current bull run in the cryptocurrency market. As the market experiences unprecedented growth and volatility, PICKLE offers a unique investment opportunity for investors looking to ride the wave of speculation and capitalize on the momentum of the market. By positioning itself as a novel and innovative investment vehicle, PICKLE aims to attract a wave of investors eager to participate in the crypto gold rush.

## **1.3 Innovative Token Distribution**

PICKLE introduces an innovative token distribution mechanism that dynamically dispenses tokens to users, or "Lil' Pickies," based on the prevailing market hype and the inflow of funds. This responsive and adaptive approach ensures a dynamic and engaging experience for users, as token rewards fluctuate in real-time according to market dynamics. By providing users with a stake in the distribution process, PICKLE fosters a sense of ownership and engagement, driving community participation and investment.

## **1.4 Promote AI-Driven Creativity**

PICKLE allocates a significant portion of its funds to the creation of AI-generated art projects, which are exclusively available for purchase using PICKLE tokens. By fostering a marketplace for AI-driven creativity, PICKLE aims to stimulate demand for the currency and promote innovation within the digital art space. Through its support of AI-driven art projects, PICKLE seeks to position itself as a pioneer in the intersection of AI, cryptocurrency, and creative expression.

# **Features and Functionality**

Pickle Rocket (PICKLE) offers a range of innovative features and functionality designed to create a dynamic and engaging platform for users, or "Lil' Pickies," to participate in the crypto revolution. These features are tailored to enhance user experience, drive community engagement, and promote the long-term growth and sustainability of the PICKLE ecosystem. These features can be divided, as the following:

### **Dynamic Token Distribution:**

- Users (Lil' Pickies) send BNB to the PICKLE contract and receive PICKLE tokens based on the current market hype.
- The dispensed tokens are influenced by the inflow of funds, ensuring a dynamic and responsive distribution mechanism.

### Staking Rewards:

- Lil' Pickies can stake their PICKLE tokens via the contract, earning bonuses in BNB and additional PICKLE tokens based on the growth of the protocol.
- 50% of the value added to the protocol is distributed to stakers, allocated as bonuses, providing an incentive for long-term engagement and participation.

### Liquidity Provisions and AI Art Projects:

- 50% of the funds collected through token distribution are used to provide liquidity to exchanges, as well as fund the creation of AI-based art projects, airdrops, and marketing materials.
- These projects will be sold on the market exclusively for PICKLE tokens, creating demand and value for the currency.

### Marketing and Community Engagement:

- PICKLE incentivizes community engagement through airdrops and mini rocket NFT airdrops.
- These initiatives give participants the opportunity to participate in closed PICKLE auctions and further contribute to the growth of the protocol.



Figure 1: **Our Fund Distribution and Tokenomics** describes how the total pool of accumulated payments to the protocol are used, and can be described by three funds: (1) *Direct Staking Rewards* - 50%, these are given directly back to stakers through fully on-chain staking mechanism, (2) *Liquidity Provision and AI Generated Content Fund* - 30%, which are used to add Liquidity to DEXs and fund AI models, and (3) *Marketing and Community Engagement Fund* - 20%, which funds marketing campaigns, NFTs, and Airdrops.

## 1.5 Dynamic Token Distribution

At the heart of PICKLE's functionality lies its innovative token distribution mechanism, which dynamically dispenses tokens to Lil' Pickies based on the prevailing market hype and the inflow of funds. This responsive and adaptive approach ensures that token rewards fluctuate in real-time according to market dynamics, providing users with an engaging and interactive experience. By offering users a stake in the distribution process, PICKLE fosters a sense of ownership and engagement, driving community participation and investment. See below for equations:

$$\text{estEpochInflow} = \left( \frac{(\text{estEpochInflow} \times 9)}{10} \right) + \left( \frac{\left( \frac{\text{amountBNB}}{(\text{block.timestamp} - \text{lastMintTime})} \right)}{10} \right)$$

$$\text{mintingAmount} = \left( \frac{\text{amountBNB}}{1 \times 10^{11}} \right) \times (\max(\text{estEpochInflow}, N)) : \text{where } N \text{ is a small constant} = 1e7$$

$$\text{tokensToMint} = \left( \frac{(\text{mintingAmount} \times N)}{\max(\text{estEpochInflow}, N)} \right) \times 1 \times 10^{11}$$

## 1.6 Staking Rewards

PICKLE introduces a revolutionary staking protocol that enables Lil' Pickies to stake their PICKLE tokens via the contract, earning bonuses in BNB and additional PICKLE tokens as the protocol grows. This incentivizes long-term engagement and participation, as a significant portion of the value added to the protocol through staking is reinvested back into the community. By rewarding users for their contribution to the ecosystem, PICKLE encourages sustained investment and loyalty, ensuring the continued growth and success of the platform. These rewards are given by a dividend share payment, that can be calculated for all stakers. See below for equations:

$$\begin{aligned} \text{stakedShare} &= \left( \frac{(\text{currDividendPool} - \text{lastDividendPool}) \times \text{amountStaked}}{\text{totalStaked}} \right) \\ \text{bonusTokens} &= \begin{cases} 0 & \text{if stakingDuration} < 60 \text{ days} \\ \text{amountStaked}/400 & \text{if } 60 \text{ days} \leq \text{stakingDuration} < 90 \text{ days} \\ \text{amountStaked}/200 & \text{if } 90 \text{ days} \leq \text{stakingDuration} < 120 \text{ days} \\ \text{amountStaked}/100 & \text{if stakingDuration} \geq 120 \text{ days} \end{cases} \\ \text{dividendShare} &= \text{stakedShare} + \left( \frac{\text{bonusTokens} \times \text{stakedShare}}{\text{totalOutstandingSupply}} \right) \end{aligned}$$

## 1.7 Liquidity Provisions and AI Art Projects

A key feature of PICKLE is its support for AI-generated art projects, which are exclusively available for purchase using PICKLE tokens. PICKLE allocates a significant portion of its funds to the creation of these art projects, which are designed to stimulate demand for the currency and promote innovation within the digital art space. By fostering a marketplace for AI-driven creativity, PICKLE aims to position itself as a pioneer in the intersection of AI, cryptocurrency, and creative expression.

## 1.8 Marketing and Community Engagement

PICKLE incentivizes community engagement through a series of initiatives, including airdrops, mini rocket NFT airdrops, and community-driven events. These initiatives provide Lil' Pickies with exclusive opportunities to participate in closed PICKLE auctions, earn rewards, and contribute to the growth of the protocol. By fostering a strong sense of community ownership and participation, PICKLE aims to cultivate a vibrant and engaged community of users, ensuring the continued success and sustainability of the platform.

## 1.9 Transparent and Secure Infrastructure

Central to PICKLE's functionality is its transparent and secure blockchain infrastructure, which ensures the integrity of all transactions and operations. By leveraging blockchain technology, PICKLE provides users with confidence in the reliability and security of the platform, fostering trust and credibility within the community. Additionally, PICKLE utilizes AI technology to optimize token distribution and enhance security measures, further enhancing the user experience and safeguarding the platform against potential threats.

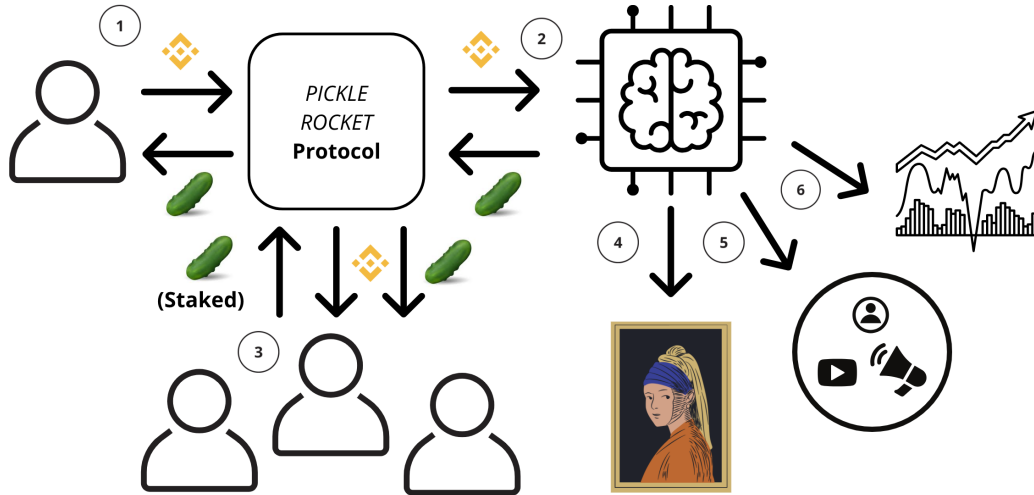


Figure 2: **Pickle Rocket Protocol Diagram** can be described as six steps: (1) user initially mints PICKLE by sending BNB, (2) AI burns PICKLE to get BNB to pay for model subscriptions and development, (3) users who have staked get rewards from protocol (See Section 1.6), (4) AI Art and NFTs are generated and auctioned to users, (5) AI marketing and advertising is generated, (6) AI analytics and improvement strategy is computed to maximize token value.

## Target Audience

Pickle Rocket (PICKLE) is designed to appeal to a diverse range of individuals within the cryptocurrency community, as well as those interested in artificial intelligence (AI) technology and digital innovation. The target audience for PICKLE includes, but is not limited to, the following segments:

### 1.10 Crypto Enthusiasts

PICKLE targets crypto enthusiasts who are actively engaged in the cryptocurrency market and seek novel investment opportunities. These individuals are familiar with blockchain technology, digital assets, and the dynamics of the crypto market. PICKLE appeals to crypto enthusiasts by offering a unique combination of AI technology, meme culture, and innovative tokenomics, providing them with an exciting and engaging investment opportunity.

### 1.11 Investors

PICKLE aims to attract investors looking to capitalize on the current bull run in the cryptocurrency market. These investors are interested in speculative assets with high growth potential and are willing to take calculated risks to maximize their returns. PICKLE offers investors a dynamic and innovative investment opportunity, with features such as dynamic token distribution, staking rewards, and AI-generated art projects, designed to generate hype and drive value appreciation.

### 1.12 Tech-Savvy Individuals

PICKLE targets tech-savvy individuals who are interested in exploring the intersection of AI technology and cryptocurrency. These individuals are fascinated by emerging technologies and are eager to experiment with new platforms and applications. PICKLE appeals to tech-savvy individuals by leveraging AI technology to optimize token distribution, enhance security measures, and facilitate the creation of AI-generated art projects, providing them with a unique and innovative platform for exploration and investment.

### 1.13 Creative Professionals

PICKLE aims to attract creative professionals, such as artists, designers, and content creators, who are interested in exploring new avenues for creative expression and monetization. These individuals are drawn to the concept of AI-generated art and are intrigued by the potential of blockchain technology to revolutionize the creative industries. PICKLE offers creative professionals a platform to showcase their talents, collaborate with other artists, and monetize their work through the creation and sale of AI-generated art projects.

### 1.14 Early Adopters

PICKLE targets early adopters who are eager to explore new technologies and platforms before they become mainstream. These individuals are willing to take risks and invest in speculative assets with the potential for high returns. PICKLE appeals to early adopters by offering exclusive

opportunities for participation, such as airdrops, mini rocket NFT airdrops, and community-driven events, providing them with a sense of belonging and ownership within the PICKLE ecosystem.

In conclusion, Pickle Rocket (PICKLE) targets a diverse range of individuals within the cryptocurrency community, as well as those interested in AI technology and digital innovation. By appealing to crypto enthusiasts, investors, tech-savvy individuals, creative professionals, and early adopters, PICKLE aims to create a vibrant and engaged community of users, driving adoption, growth, and value appreciation within the PICKLE ecosystem.

## **Other Technical Aspects**

Pickle Rocket (PICKLE) leverages advanced blockchain technology and artificial intelligence (AI) algorithms to create a secure, transparent, and innovative platform for users to participate in the crypto revolution. The technical aspects of PICKLE encompass various components, including blockchain infrastructure, smart contract functionality, AI integration, and security measures.

### **1.15 Blockchain Infrastructure**

PICKLE operates on a decentralized blockchain network, providing users with a secure and transparent platform for conducting transactions and interacting with the protocol. By leveraging blockchain technology, PICKLE ensures the integrity of all transactions and operations, while also providing users with visibility into the transaction history and token distribution process. The decentralized nature of the blockchain network enhances security, eliminates single points of failure, and fosters trust and credibility within the community.

### **1.16 Smart Contract Functionality**

Central to PICKLE's functionality is its use of smart contracts, which are self-executing contracts with the terms of the agreement directly written into code. PICKLE utilizes smart contracts to automate various processes within the protocol, including token distribution, staking rewards, and AI-generated art projects. Smart contracts ensure the reliability and efficiency of these processes, while also providing users with transparency and immutability. Additionally, smart contracts enable PICKLE to implement innovative features, such as dynamic token distribution and staking bonuses, enhancing the user experience and driving community engagement.

### **1.17 AI Integration**

PICKLE integrates artificial intelligence (AI) algorithms to optimize token distribution, enhance security measures, and facilitate the creation of AI-generated art projects. AI technology is used to analyze market data and sentiment, allowing PICKLE to dynamically adjust token rewards based on the prevailing market hype and the inflow of funds. Additionally, AI algorithms are employed to identify and mitigate potential security threats, ensuring the integrity of the platform and protecting user assets. Furthermore, AI-driven algorithms are utilized to generate unique and innovative art projects, stimulating demand for PICKLE tokens and fostering creativity within the digital art space.



## 1.18 Security Measures

PICKLE implements robust security measures to protect user assets and ensure the integrity of the platform. In addition to leveraging blockchain technology for decentralized security, PICKLE employs various security protocols, such as encryption, multi-factor authentication, and auditing mechanisms [1]. These security measures help safeguard user funds, prevent unauthorized access, and mitigate potential security vulnerabilities. Furthermore, PICKLE conducts regular security audits and penetration testing to identify and address any potential security risks, ensuring the continued safety and reliability of the platform.

In conclusion, Pickle Rocket (PICKLE) utilizes advanced blockchain technology and artificial intelligence (AI) algorithms to create a secure, transparent, and innovative platform for users to participate in the crypto revolution. By leveraging blockchain infrastructure, smart contract functionality, AI integration, and security measures, PICKLE aims to provide users with a dynamic and engaging experience, driving adoption, growth, and value appreciation within the PICKLE ecosystem.

## 2 Learning by Example - "Shitcoins"

### 2.1 The Good

While the cryptocurrency market has seen its fair share of failures and scams, there are also projects that have gained significant traction and community support. Here are some examples of cryptocurrencies that have achieved success [2]:

1. Dogecoin (DOGE): Dogecoin was created in 2013 by software engineers Billy Markus and Jackson Palmer. It was inspired by the meme of a Shiba Inu dog and was intended to be a joke cryptocurrency to attract mainstream attention. As a fork of Litecoin (LTC), DOGE adopts the same Proof of Work (POW) mechanism, and it has no maximum supply. Dogecoin has achieved remarkable success, with a current ROI of over 100,000% and a market capitalization exceeding \$30 billion, making it one of the top cryptocurrencies by market cap [6].
2. Shiba Inu (SHIB): Shiba Inu (SHIB) is the rival of DOGE and is often referred to as the "Dogecoin killer". SHIB is also named after a Japanese dog breed. It was created by an anonymous developer named Ryoshi in August 2020. The main difference between DOGE and SHIB is that the latter has a limited supply of 1 quadrillion tokens, of which 50% were burnt and donated to charity. SHIB's ecosystem also includes a decentralized exchange, an NFT art incubator, NFTs, and an NFT game. SHIB has gained significant popularity, with a current ROI of over 1,000,000% and a market capitalization in the billions.
3. DogWifHat (WIF): DogWifHat, also known as Wifey, is a cryptocurrency that gained attention for its unique branding and community-driven approach. It has built a loyal following and continues to see steady growth in adoption. While its ROI and market capitalization may not be as high as some other cryptocurrencies, it has maintained a strong community and stable performance.
4. SafeMoon (SAFEMOON): SafeMoon is a BEP-20 token launched on the BNB Smart Chain (BSC) in March 2021. SAFEMOON rewards long-term holders by penalizing those who sell the token with a 10% exit fee, of which half of the fees will be distributed to existing

SAFEMOON holders, and the other half will be burnt. It attracted retail investors' attention after it soared in April. As of November 2021, SAFEMOON has a 9418.54% ROI, according to CoinMarketCap, and a market capitalization in the billions.

## 2.2 The Bad

Below is a list of less reputable or unsuccessful cryptocurrencies, commonly referred to as "shitcoins," that have faced criticism or failure in the cryptocurrency market:

1. BitConnect (BCC): BitConnect was a cryptocurrency and investment platform that was widely criticized as a Ponzi scheme. It collapsed in 2018, resulting in significant financial losses for investors.
2. OneCoin (ONE): OneCoin was another cryptocurrency project that faced allegations of being a Ponzi scheme. Its founder, Ruja Ignatova, disappeared in 2017, and the project's operations have since been shut down by law enforcement agencies.
3. Prodeum (PDE): Prodeum was a cryptocurrency project that aimed to create a platform for tracking fruits and vegetables on the blockchain. However, it turned out to be a scam, with the project's website disappearing shortly after its launch.
4. Bitpetite (BTP): Bitpetite was a high-yield investment program that promised guaranteed returns on cryptocurrency investments. It was later revealed to be a Ponzi scheme, and its operators disappeared with investors' funds.
5. Confido (CFD): Confido was a cryptocurrency project that aimed to create a platform for secure peer-to-peer transactions. However, the project's founders disappeared shortly after raising funds through an initial coin offering (ICO), leading to accusations of fraud.

It is important to learn from the mistakes and failures of these projects to avoid similar pitfalls and ensure the success and sustainability of Pickle Rocket (PICKLE).

## 2.3 The Ugly

In addition to the successes and failures of individual cryptocurrencies, it is also important to consider the broader context of currency systems and monetary history. While traditional fiat currencies may not be considered "shitcoins" in the same sense as cryptocurrencies, they have their own shortcomings and limitations. Here are some examples:

1. United States Dollar (USD): The US dollar is the world's primary reserve currency and is widely used for international trade. However, it is also subject to inflation and government manipulation, leading to concerns about its long-term stability.
2. Central Bank Digital Currencies (CBDC): CBDCs are digital currencies issued by central banks, such as the proposed digital yuan by the People's Bank of China. While they offer benefits in terms of efficiency and transparency, they restrict privacy and enforce government controls over financial transactions.

3. Venezuelan Bolivar Fuerte: The Venezuelan bolivar has experienced hyperinflation in recent years, leading to economic instability and widespread poverty. The government's attempts to stabilize the currency have been largely unsuccessful, resulting in the adoption of alternative currencies such as Bitcoin [5].
4. German Papiermark: The German Papiermark was the currency of the Weimar Republic during the hyperinflation of the early 1920s. It became virtually worthless, leading to social unrest and political upheaval [4].
5. Sea Shells: Before the advent of modern currency systems, various societies used commodities such as sea shells as a medium of exchange. While they served as a form of currency, they lacked the stability and durability of modern currencies. When large portions of shells were brought by international trade it destabilized much of the native economies.

## Conclusion

Pickle Rocket (PICKLE) represents a novel approach to meme coins, combining elements of AI technology with cryptocurrency to create a dynamic and engaging platform for users. With its innovative token distribution mechanism, staking rewards, and AI-driven art projects, PICKLE aims to capture the imagination of the crypto community and propel itself to the moon.

## References

- [1] Binance. *Binance Coin Whitepaper*. Year. URL: <https://www.exodus.com/assets/docs/binance-coin-whitepaper.pdf> (visited on 03/09/2024).
- [2] Binance Academy. *What are Meme Coins?* Year. URL: [https://academy.binance.com/en/articles/what-are-meme-coins?utm\\_campaign=googleadsxacademy&utm\\_source=googleadwords\\_int&utm\\_medium=cpc&ref=HDYAHEES&gad\\_source=1&gclid=EAiaIQobChMI9-vfn8PohAMVPWJIAB20hQPYEAAAYASAAEgJe4PD\\_BwE](https://academy.binance.com/en/articles/what-are-meme-coins?utm_campaign=googleadsxacademy&utm_source=googleadwords_int&utm_medium=cpc&ref=HDYAHEES&gad_source=1&gclid=EAiaIQobChMI9-vfn8PohAMVPWJIAB20hQPYEAAAYASAAEgJe4PD_BwE) (visited on 03/09/2024).
- [3] Adedoyin A. Hussain and Fadi Al-Turjman. "Artificial intelligence and blockchain: A review". In: *Transactions on Emerging Telecommunications Technologies* Volume Number.Issue Number (2021), Page Range. DOI: 10.1002/ett.4268. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.4268> (visited on 03/09/2024).
- [4] *Hyperinflation in the Weimar Republic*. Year. URL: [https://en.wikipedia.org/wiki/Hyperinflation\\_in\\_the\\_Weimar\\_Republic](https://en.wikipedia.org/wiki/Hyperinflation_in_the_Weimar_Republic) (visited on 03/09/2024).
- [5] *Hyperinflation in Venezuela*. Year. URL: [https://en.wikipedia.org/wiki/Hyperinflation\\_in\\_Venezuela](https://en.wikipedia.org/wiki/Hyperinflation_in_Venezuela) (visited on 03/09/2024).
- [6] Billy Markus and Jackson Palmer. *Dogecoin: A Peer-to-Peer Digital Currency*. 2013. URL: <https://dogechain.dog/DogechainWP.pdf> (visited on 03/09/2024).
- [7] Satoshi Nakamoto. *Bitcoin: A Peer-to-Peer Electronic Cash System*. 2008. URL: <https://bitcoin.org/bitcoin.pdf> (visited on 03/09/2024).

### 3 Appendix

The code is open sourced and documented below:

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

// Importing OpenZeppelin contracts for ERC20 functionality, ownership,
// burning tokens, security against reentrancy attacks, and mathematical
// operations.
import "@openzeppelin/contracts/token/ERC20/extensions/ERC20Capped.sol";
import "@openzeppelin/contracts/access/Ownable.sol";
import "@openzeppelin/contracts/token/ERC20/extensions/ERC20Burnable.sol";
import "@openzeppelin/contracts/security/ReentrancyGuard.sol";
import "@openzeppelin/contracts/utils/math/Math.sol";

// PickleRocket contract inherits from ERC20, ERC20Capped for maximum token
// supply, Ownable for admin control, ERC20Burnable for burning tokens,
// and ReentrancyGuard for security.
contract PickleRocket is ERC20, ERC20Capped, Ownable, ERC20Burnable,
    ReentrancyGuard {
    // Private constant for the token cap, set to 1 billion tokens,
    // considering the 18 decimal places.
    uint256 private constant _cap = 1000000000 * 10**18;

    // Public variables for tracking contract states.
    address public developerWallet; // Wallet address for the developer.
    uint256 public lastDividendPaymentTime; // Timestamp of the last
    // dividend payment.
    uint256 public totalStaked; // Total amount of tokens staked by users.
    uint256 public dividendPool; // Total amount of ETH in the dividend
    // pool.
    uint256 public estEpochInflow; // Estimated inflow of tokens per epoch.
    uint256 public accumulatedMintedCoins; // Total amount of coins minted
    // so far.
    uint256 public lastMintTime = 0; // Timestamp of the last minting event
    // .
    uint256 private N = 1e7; // A normalization constant for calculations.
    uint256 public stallEndTime; // Timestamp indicating the end of a stall
    // period for developer funds access.
    uint256 public minimumStake = 1000; // Minimum amount required to stake
    // , set to 1 million tokens.

    // Mapping for tracking staked amounts and staking timestamps for each
    // address.
    mapping(address => Stake) public stakes;
    // Mapping to check if an address is currently staking.
    mapping(address => bool) private isStaker;
    // Dynamic array for keeping track of all stakers.
    address[] private stakerList;
    // Mapping for tracking unclaimed dividends for each address.
```

```

mapping(address => uint256) public unclaimedDividends;

// Struct to hold staking details: amount staked and timestamp of
// staking.
struct Stake {
    uint256 amount;
    uint256 timestamp;
    uint256 initDividendPool;
}

// Events for logging activities on the blockchain.
event Staked(address indexed staker, uint256 amount, uint256 timestamp)
    ;
event Unstaked(address indexed staker, uint256 amount, uint256
    timestamp);
event DividendsDistributed(uint256 totalDividend);
event DividendsClaimed(address indexed staker, uint256 amount);

// Constructor to set initial values for the contract upon deployment.
constructor(address _developerWallet)
    Ownable(msg.sender) // Set contract deployer as the owner.
    ERC20("Pickle_Rocket", "PICKLE") // Set token name and symbol.
    ERC20Capped(_cap) // Set the maximum cap for the token supply.
{
    // Ensuring the developer wallet address is not the zero address.
    require(_developerWallet != address(0), "Developer_wallet_cannot_be
        _the_zero_address");

    // Initializing contract variables.
    developerWallet = _developerWallet;
    _mint(_developerWallet, 100000000 * 10**18); // Minting initial
        tokens to the developer wallet.
    lastMintTime = block.timestamp; // Setting the last mint time to
        the current block timestamp.
    estEpochInflow = 0; // Initializing estimated epoch inflow to zero.
    accumulatedMintedCoins = 0; // Initializing accumulated minted
        coins to zero.
    dividendPool = 0; // Initializing dividend pool to zero.
    lastDividendPaymentTime = block.timestamp; // Setting the last
        dividend payment time to the current block timestamp.
    stallEndTime = block.timestamp; // Initializing stall end time to
        the current block timestamp.
}

// Override function to ensure compatibility between ERC20 and ERC20Capped.
function _update(address from, address to, uint256 value) internal
    override(ERC20, ERC20Capped) {
    super._update(from, to, value);
}

// Receive function to handle direct ETH payments to the contract.

```

```

receive() external payable nonReentrant {
    // Calculating the share of the incoming ETH to add to the dividend
    // pool.
    uint256 dividendShare = msg.value / 2;
    dividendPool += dividendPool + dividendShare;
    // Updating the estimated epoch inflow with the new transaction
    // value.
    estEpochInflow = ((estEpochInflow * 9) / 10) + ((msg.value / (block
    .timestamp - lastMintTime)) / 10);

    // Calculating the amount of tokens to mint based on the ETH sent
    // minus the dividend share.
    uint256 mintingAmount = msg.value - dividendShare;
    // Formula to determine the number of tokens to mint, considering
    // the estimated epoch inflow.
    uint256 tokensToMint = ((mintingAmount * N) / Math.max(
    estEpochInflow, N)) * 1e11;
    // Ensuring minting does not exceed the cap.
    require(totalSupply() + tokensToMint <= _cap, "Minting would exceed
    cap");
    accumulatedMintedCoins += tokensToMint;
    _mint(msg.sender, tokensToMint); // Minting tokens to the sender.
    lastMintTime = block.timestamp; // Updating the last mint time.
}

// Function for users to stake their tokens.
function stake(uint256 _amount) public nonReentrant {
    // Checking if the staking amount meets the minimum requirement.
    require(_amount >= minimumStake, "Amount must be at least the
    minimum stake");
    // Ensuring the caller has enough tokens to stake.
    require(balanceOf(msg.sender) >= _amount, "Insufficient token
    balance");

    // Transferring the tokens from the caller to the contract for
    // staking.
    _transfer(msg.sender, address(this), _amount);

    // Updating the staking details for the caller.
    if (stakes[msg.sender].amount > 0) {
        stakes[msg.sender].amount += _amount;
        stakes[msg.sender].timestamp = block.timestamp;
        stakes[msg.sender].initDividendPool = dividendPool;
    } else {
        stakes[msg.sender] = Stake(_amount, block.timestamp,
        dividendPool);
        isStaker[msg.sender] = true;
        stakerList.push(msg.sender);
    }
}

```

```

    // Updating the total staked amount.
    totalStaked += _amount;
    // Emitting an event for the staking action.
    emit Staked(msg.sender, _amount, block.timestamp);
}

// Function for users to withdraw their dividends.
function withdrawDividends() public nonReentrant {
    // Ensuring the caller has staked tokens.
    require(stakes[msg.sender].amount > 0, "You have no staked tokens");
    ;
    // Ensuring the staking duration meets the 30-day minimum
    requirement.
    require(block.timestamp - stakes[msg.sender].timestamp >= 30 days,
        "30-day minimum staking required"); // TODO
    uint256 stakingDuration = block.timestamp - stakes[msg.sender].
        timestamp;
    // Calculating the user's share of the dividend based on their
    staked amount.
    uint256 dividendShare = ((dividendPool - stakes[msg.sender].
        initDividendPool) * stakes[msg.sender].amount) / totalStaked;

    // Calculating bonus tokens based on the duration of staking.
    uint256 bonusTokens = 0;
    if (stakingDuration >= 60 days && stakingDuration < 90 days) {
        bonusTokens = stakes[msg.sender].amount / 400;
    } else if (stakingDuration >= 90 days && stakingDuration < 120 days
        ) {
        bonusTokens = stakes[msg.sender].amount / 200;
    } else if (stakingDuration >= 120 days) {
        bonusTokens = stakes[msg.sender].amount / 100;
    }

    // Adjusting the dividend share to include any bonus tokens.
    dividendShare = dividendShare + ((bonusTokens * dividendPool) /
        totalSupply());

    // Resetting the caller's unclaimed dividends to zero and
    transferring the dividend share.
    payable(msg.sender).transfer(dividendShare);
    _mint(msg.sender, bonusTokens); // Minting bonus tokens to the
    caller.

    stakes[msg.sender].timestamp = block.timestamp;
    stakes[msg.sender].initDividendPool = dividendPool;

    // Emitting an event for the dividend claim.
    emit DividendsClaimed(msg.sender, dividendShare);
}

```

```

// Function for users to unstake their entire staked tokens.
function unstake() public nonReentrant {
    // Ensuring the caller is currently staking.
    require(isStaker[msg.sender], "You have no staked tokens");

    // Retrieving the total staked amount for the caller.
    uint256 stakedAmount = stakes[msg.sender].amount;

    // Ensuring there is a non-zero staked amount.
    require(stakedAmount > 0, "Insufficient staked amount");

    // Resetting the staker's stake to zero.
    stakes[msg.sender].amount = 0;
    isStaker[msg.sender] = false;

    // Updating the total staked amount.
    totalStaked -= stakedAmount;

    // Transferring the unstaked tokens back to the caller.
    _transfer(address(this), msg.sender, stakedAmount);

    // Emitting an event for the unstaking action.
    emit Unstaked(msg.sender, stakedAmount, block.timestamp);
}

// Function for the developer to access funds.
function accessDeveloperFunds(uint256 amount) public nonReentrant {
    // Ensuring the caller is the developer wallet.
    require(msg.sender == developerWallet, "Only the developer wallet can access funds");
    // Ensuring the stall period has ended.
    require(block.timestamp >= stallEndTime, "Stall period has not ended");
    // Ensuring the developer wallet has enough tokens to burn.
    require(amount <= balanceOf(developerWallet), "Insufficient balance");

    // Burning the specified amount of tokens from the developer wallet
    _burn(developerWallet, amount);

    // Calculating the ETH value equivalent to the burned tokens.
    uint256 ethValue = (amount / 1e11) * (Math.max(estEpochInflow, N) / N);
    // Ensuring there is enough ETH in the contract to transfer.
    require(ethValue <= address(this).balance, "Insufficient ETH in contract");

    // Transferring the ETH value to the developer wallet.

```



```

        (bool sent, ) = developerWallet.call{value: ethValue}("");
        require(sent, "ETH_transfer_failed");

        // Updating the stall end time to 7 days from now.
        stallEndTime = block.timestamp + 7 days;
    }

    // Function for the owner to update the developer wallet address.
    function updateDeveloperWallet(address newDeveloperWallet) public
        onlyOwner {
        // Ensuring the new developer wallet address is not the zero
        address.
        require(newDeveloperWallet != address(0), "New_developer_wallet_
        address_cannot_be_the_zero_address");
        // Updating the developer wallet address.
        developerWallet = newDeveloperWallet;
    }

    // View function to get the current balance of the dividend pool.
    function getDividendPoolBalance() public view returns (uint256) {
        return dividendPool;
    }

    // View function to get the staked amount for a specific staker.
    function getStakedAmount(address staker) public view returns (uint256)
    {
        return stakes[staker].amount;
    }

    // View function to get the timestamp amount for a specific staker.
    function getStakedTimestamp(address staker) public view returns (
        uint256) {
        return stakes[staker].timestamp;
    }

    // View function to get the dividend pool amount for a specific staker.
    function getStakedInitDividendPool(address staker) public view returns
        (uint256) {
        return stakes[staker].initDividendPool;
    }

    // View function to get the staked amount for a specific staker.
    function getTotalStaked() public view returns (uint256) {
        return totalStaked;
    }
}

```