UPALGO - An Online Coding Practice Platform

A PROJECT REPORT
for
Mini Project (KCA353)
Session (2023-24)

Submitted by
Aniket Sharma

2200290140027

Submitted in partial fulfillment of the
Requirements for the Degree of

MASTER OF COMPUTER APPLICATION

Under the Supervision of
Ms. Divya Singhal
Assistant Professor

Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206

(JANUARY 2024)

CERTIFICATE

Certified that **Aniket Sharma (2200290140027)** has/ have carried out the project work having "**UpAlgo : Coding Practice Platform**" (**Mini Project-KCA353**) for **Master of Computer Application** from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date:

Aniket Sharma
2200290140027

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

Ms. Divya Singhal	**Dr. Arun Tripathi**
Assistant Professor	**Head**
Department of Computer Applications	**Department of Computer Applications**
KIET Group of InstitutionsGhaziabad	**KIET Group of Institutions Ghaziabad**

ABSTRACT

Upalgo, a Django-backed web application, reimagines the coding challenge experience, drawing inspiration from LeetCode. Offering a curated mix of 10 algorithmic questions—5 unpaid and 5 paid—the platform prioritizes user engagement and skill enhancement. GitHub OAuth simplifies secure user logins, while Stripe integration ensures seamless and secure payments for premium content. The platform's mission is to provide coding enthusiasts with a diverse and challenging environment, fostering continuous improvement in algorithmic problem-solving skills.

ACKNOWLEDGEMENTS

Success in life is never attained single-handedly. My deepest gratitude goes to my project supervisor, **Ms. Divya Singhal** for her guidance, help, and encouragement throughout my project work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to **Dr. Arun Kumar Tripathi**, Professor and Head, Department of Computer Applications, for his insightful comments and administrative help on various occasions.

Fortunately, I have many understanding friends, who have helped me a lot in many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me with moral support and other kinds of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

List of Abbreviations

1. **Upalgo**: Up Algorithm

2. **OAuth**: Open Authorization

3. **API**: Application Programming Interface

4. **SQL**: Structured Query Language

5. Django: Web framework for Python

6. React.js: JavaScript library for building user interfaces

7. MySQL: Structured Query Language Database

8. GitHub: Web-based hosting service for version control

9. Stripe: Online payment processing for internet businesses

TABLE OF CONTENTS

	Certificate	i
	Abstract	ii
	Acknowledgements	iii
	List Of Abbreviations	iv
	Table of Content	v
1.	Introduction	1
2.	Design	7
3.	Testing	12
4.	Conclusion	14
5.	Bibliography	17

1. Introduction

Upalgo, a sophisticated web application, transcends the traditional coding challenge experience, powered by Django as its backend framework and MySQL as the database of choice. This extended section provides a more in-depth exploration of various aspects of the project, aiming for a comprehensive understanding.

1.1 Project Overview

Upalgo stands at the intersection of accessibility and challenge, offering a carefully curated collection of 10 algorithmic questions. This section delves further into the intricacies of the unpaid and paid question sets, emphasizing their role in catering to a diverse audience of coding enthusiasts.

1.1.1 Unpaid and Paid Questions

The Unpaid and Paid question sets serve as the heartbeat of Upalgo, strategically designed to meet the preferences of users at different stages of their coding journey. This subsection elucidates the rationale behind the selection process, ensuring a balanced mix of difficulty levels and problem-solving paradigms.

1.1.2 GitHub OAuth for Seamless Authentication

User authentication is a pivotal element of any web application. Upalgo goes the extra mile by incorporating GitHub OAuth, offering users a familiar and secure login experience. This subsection explores the implementation intricacies, emphasizing the user-centric design choices made for a seamless onboarding process.

1.1.3 Test Cases for Thorough Evaluation

Beyond the surface level, the inclusion of 10 test cases per question reveals the commitment to user success. This subsection details the methodology behind the test case selection, emphasizing the importance of robust evaluation metrics in fostering a culture of correctness and efficiency in code implementation.

1.1.4 Payment Integration with Stripe

Monetization is an essential consideration for platform sustainability. The integration of Stripe for payment processing opens new avenues for Upalgo. This subsection delves into the decision-making process behind choosing Stripe, highlighting the importance of a secure and reliable payment gateway for both users and the platform.

1.2. Technical Details

The technical underpinnings of Upalgo lay the foundation for its functionality and user experience. This extended exploration provides deeper insights into the technology choices made, showcasing the synergy between Django, React.js, and MySQL.

1.2.1 Tech Stack

The synergy between Django, React.js, and MySQL is more than a mere technical choice; it's a strategic decision to ensure a harmonious and efficient development and user experience. This subsection elaborates on the strengths of each component in the tech stack, showcasing their collective impact on Upalgo's robust architecture.

1.2.2 GitHub Repository

Beyond being a version control hub, the GitHub repository serves as a beacon of collaboration and transparency. This subsection explores the collaborative potential of an open-source project, encouraging users and developers to actively engage with the codebase, contribute enhancements, and shape the evolution of Upalgo.

1.3. Challenges and Solutions

The development journey of Upalgo was marked by challenges that demanded creative solutions. This extended exploration provides a deeper dive into the complexities faced and the innovative solutions devised to surmount them.

1.3.1 Payment Security Concerns

The landscape of online transactions demands a meticulous approach to security. This subsection goes beyond the surface, exploring the specific security concerns addressed during the integration of the Stripe API. It sheds light on the encryption protocols, data handling practices, and the overall commitment to fortifying the platform against potential threats.

1.3.2 Balancing Question Difficulty

Curating a question set that caters to users with diverse skill levels is an art in itself. This subsection takes a magnifying glass to the process, discussing the criteria used for question selection, the feedback loops established for continuous improvement, and the delicate balance struck to ensure a rewarding experience for users of all proficiencies.

1.4. Future Roadmap

Upalgo's journey extends far beyond its current state. The future roadmap envisions growth, evolution, and a deepened connection with its user community.

1.4.1 User Analytics Implementation

The incorporation of user analytics isn't just a technical enhancement; it's a strategic move to align the platform with user needs. This subsection details the specific metrics tracked, the tools employed for analysis, and the envisioned impact on refining Upalgo based on real user interactions.

1.4.2 Community Features Integration

Community building is a cornerstone of Upalgo's future plans. This subsection explores the potential of discussion forums, leaderboards, and user-generated content in creating a thriving community. It delves into the user engagement strategies, moderation mechanisms, and the envisioned role of the community in shaping the platform's trajectory.

Conclusion

In a grand synthesis of technology, user experience, and community engagement, Upalgo emerges as a beacon in the coding challenge landscape. This extended conclusion wraps up the introductory section, emphasizing the overarching vision of Upalgo as a catalyst for continuous learning, collaboration, and skill development in the realm of algorithmic problem-solving.

2. Design: Upalgo Project

The design phase of the Upalgo project encompasses the architectural, user experience, and database design considerations, ensuring a cohesive and user-friendly platform. This section delves into the various facets of the design process.

2.1. Architectural Design

The architectural design of Upalgo is structured to ensure scalability, flexibility, and optimal performance. The Django framework serves as the backbone, providing a robust backend that seamlessly interacts with the React.js frontend. This decoupled architecture allows for the independent development and scalability of each component, enhancing the overall maintainability of the system.

2.1.1 Backend Design

The backend architecture follows the Model-View-Controller (MVC) pattern inherent in Django. Models define the data structure, views handle user interface logic, and controllers manage the flow of data between the model and view. This separation of concerns simplifies development, making code modular and enhancing code readability.

2.1.2 Frontend Design

React.js, the chosen frontend library, facilitates a component-based approach to UI design. The modular structure of React components aligns with the backend's MVC pattern, promoting code reusability and maintainability. The use of state management ensures a dynamic and responsive user interface, crucial for an engaging coding challenge platform.

2.1.3. User Experience (UX) Design

The user experience design of Upalgo is crafted to prioritize user engagement, accessibility, and simplicity.

2.1.4 GitHub OAuth Integration

The integration of GitHub OAuth for user authentication ensures a familiar and secure login experience. This strategic choice not only simplifies onboarding but also establishes trust, leveraging the security features inherent in GitHub authentication.

2.2 Question Set Presentation

The presentation of the question sets—unpaid and paid—is designed to be intuitive and visually appealing. Clear categorization and user-friendly interfaces guide users through the selection process, promoting an engaging and efficient user experience.

2.3 Test Case Interaction

The interaction with test cases is designed to be seamless, allowing users to validate their solutions comprehensively. Intuitive controls and real-time feedback enhance the learning experience, encouraging users to iterate on their solutions until they meet the specified criteria.

2.4 Payment Process Flow

For users opting for premium content, the payment process flow is designed with simplicity and security in mind. Stripe integration seamlessly guides users through the payment process, ensuring a trustworthy and efficient transaction experience.

2.5. Database Design

The database design of Upalgo, implemented using MySQL, focuses on data integrity, normalization, and efficient retrieval.

2.5.1 Question and Test Case Storage

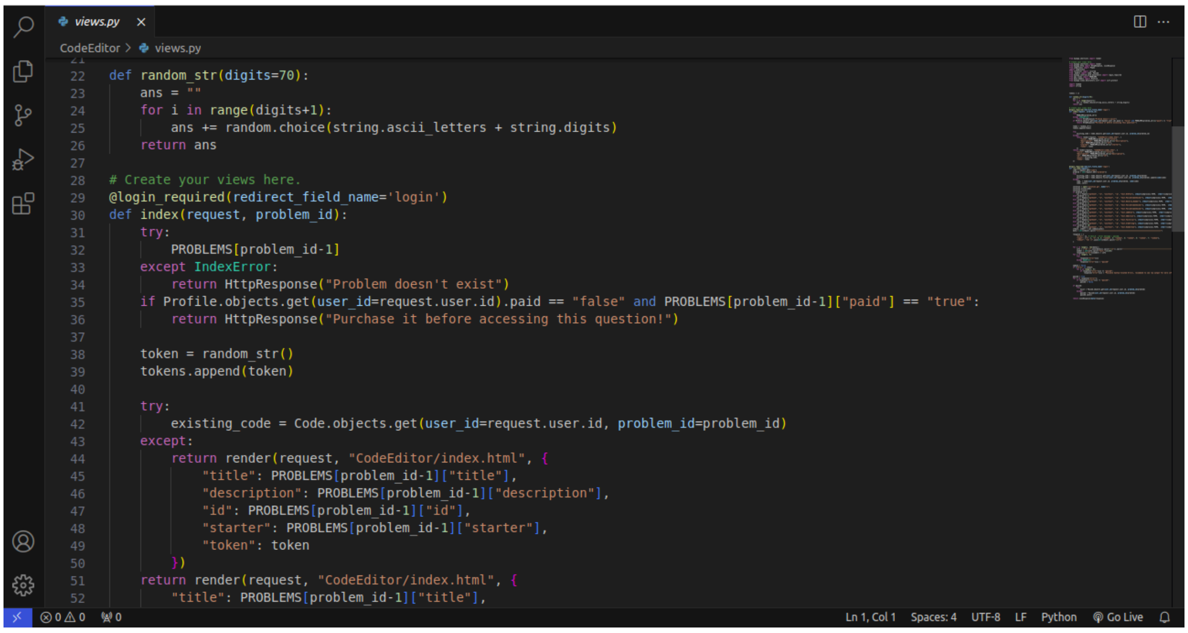
Questions and their associated test cases are stored in a structured manner, optimizing for quick retrieval during user interactions. Normalization techniques are applied to minimize redundancy and ensure data consistency.

2.6 User Profile and Authentication

User profiles and authentication data are securely stored, leveraging encryption and hashing techniques to safeguard sensitive information. This design choice aligns with the project's commitment to user privacy and data security.

2.7. Conclusion

The design phase of Upalgo lays the groundwork for a robust, user-centric, and scalable platform. The architectural decisions ensure a flexible and maintainable codebase, while the user experience and database designs prioritize engagement, simplicity, and data integrity. As Upalgo progresses from design to implementation, these considerations form the bedrock of a platform poised for success in the competitive coding challenge landscape.

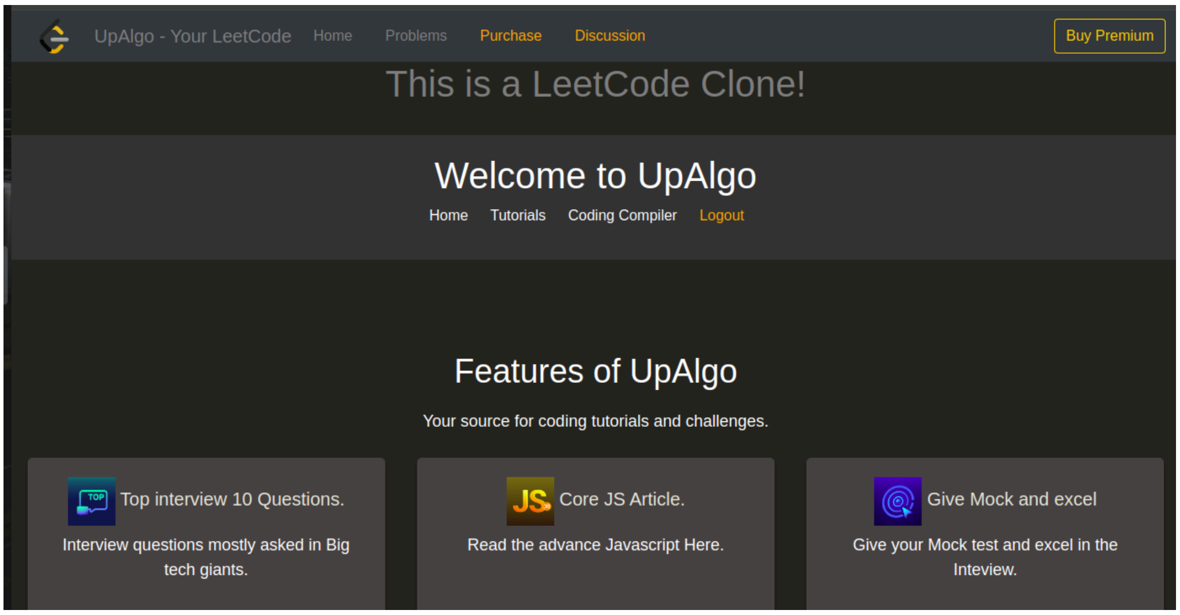


```python
def random_str(digits=70):
    ans = ""
    for i in range(digits+1):
        ans += random.choice(string.ascii_letters + string.digits)
    return ans

# Create your views here.
@login_required(redirect_field_name='login')
def index(request, problem_id):
    try:
        PROBLEMS[problem_id-1]
    except IndexError:
        return HttpResponse("Problem doesn't exist")
    if Profile.objects.get(user_id=request.user.id).paid == "false" and PROBLEMS[problem_id-1]["paid"] == "true":
        return HttpResponse("Purchase it before accessing this question!")

    token = random_str()
    tokens.append(token)

    try:
        existing_code = Code.objects.get(user_id=request.user.id, problem_id=problem_id)
    except:
        return render(request, "CodeEditor/index.html", {
            "title": PROBLEMS[problem_id-1]["title"],
            "description": PROBLEMS[problem_id-1]["description"],
            "id": PROBLEMS[problem_id-1]["id"],
            "starter": PROBLEMS[problem_id-1]["starter"],
            "token": token
        })
    return render(request, "CodeEditor/index.html", {
        "title": PROBLEMS[problem_id-1]["title"],
```



This is a LeetCode Clone!

Welcome to UpAlgo

Features of UpAlgo

Your source for coding tutorials and challenges.

Top interview 10 Questions.

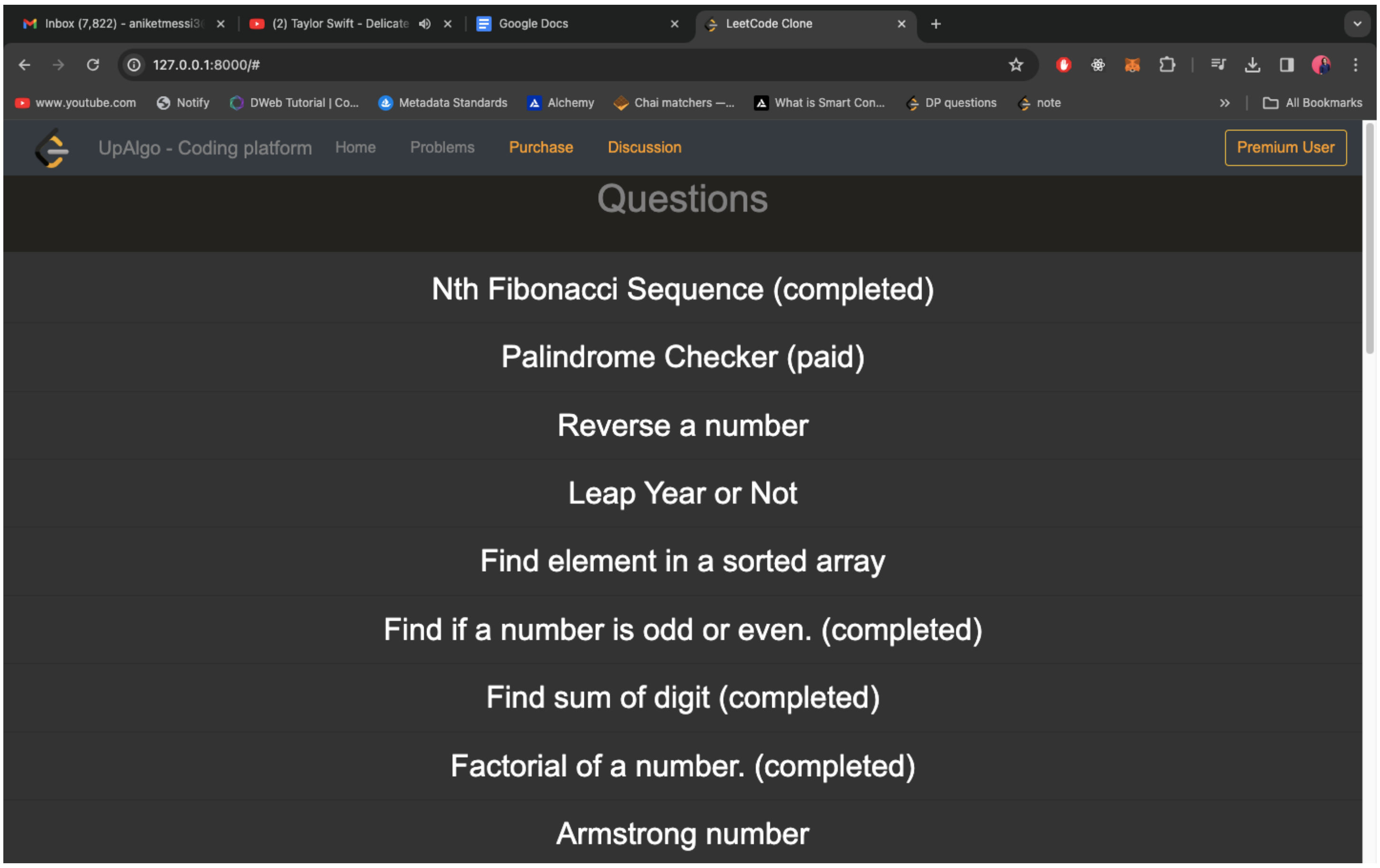
Interview questions mostly asked in Big tech giants.

Core JS Article.

Read the advance Javascript Here.

Give Mock and excel

Give your Mock test and excel in the Inteview.



Questions

Nth Fibonacci Sequence (completed)

Palindrome Checker (paid)

Reverse a number

Leap Year or Not

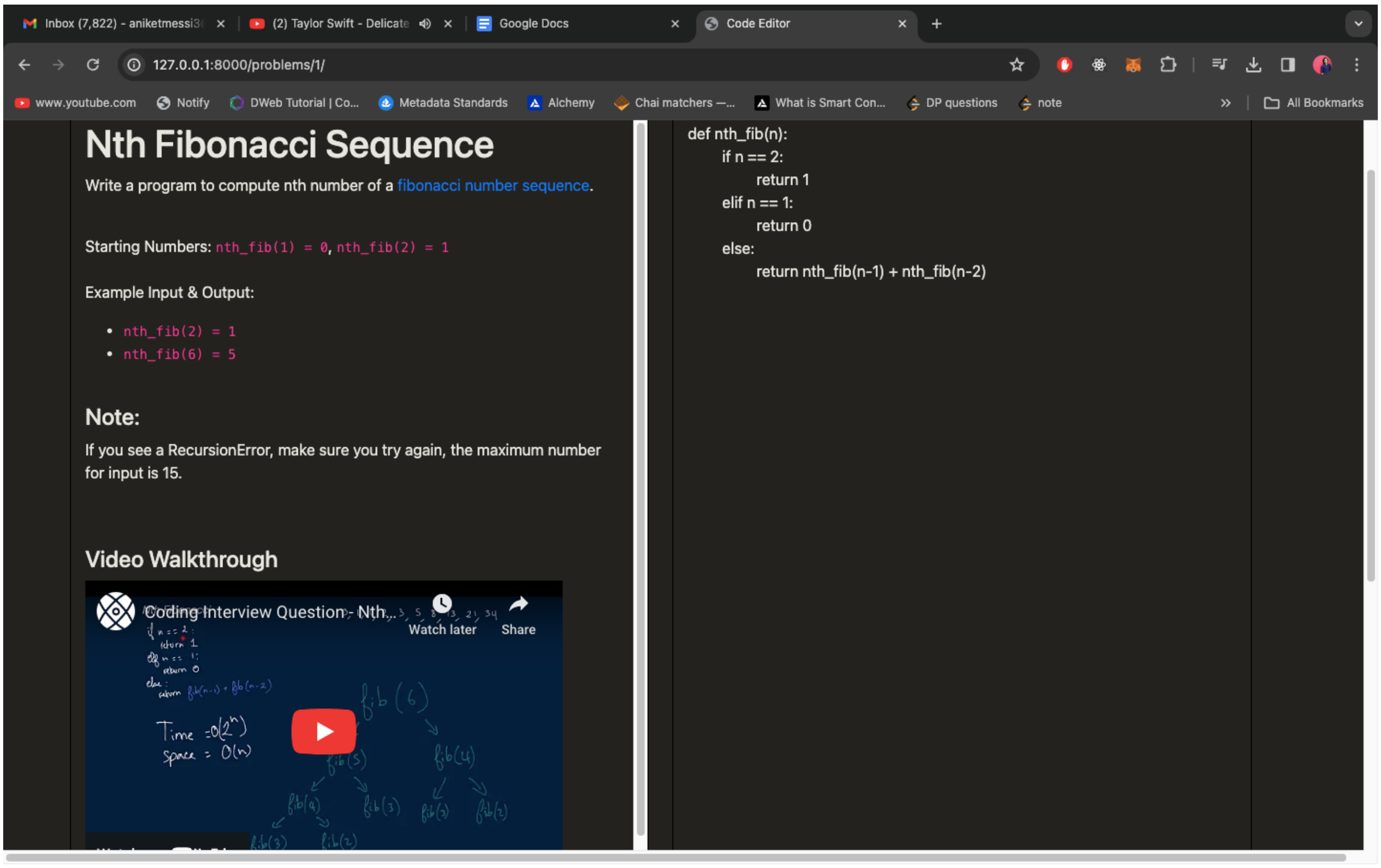
Find element in a sorted array

Find if a number is odd or even. (completed)

Find sum of digit (completed)

Factorial of a number. (completed)

Armstrong number



Nth Fibonacci Sequence

Write a program to compute nth number of a fibonacci number sequence.

Starting Numbers: `nth_fib(1) = 0`, `nth_fib(2) = 1`

Example Input & Output:

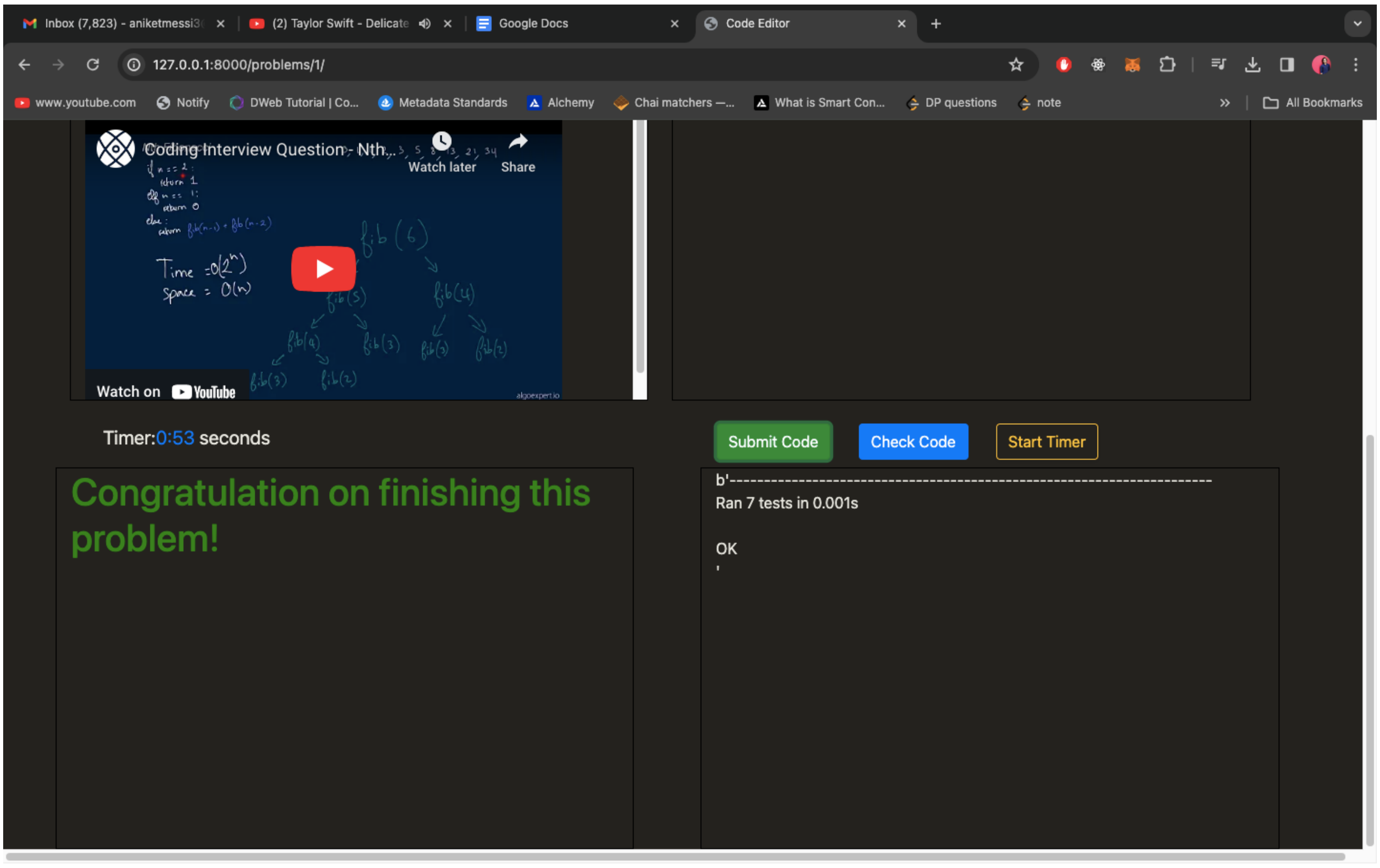
- `nth_fib(2) = 1`
- `nth_fib(6) = 5`

Note:

If you see a RecursionError, make sure you try again, the maximum number for input is 15.

Video Walkthrough

```
def nth_fib(n):
    if n == 2:
        return 1
    elif n == 1:
        return 0
    else:
        return nth_fib(n-1) + nth_fib(n-2)
```

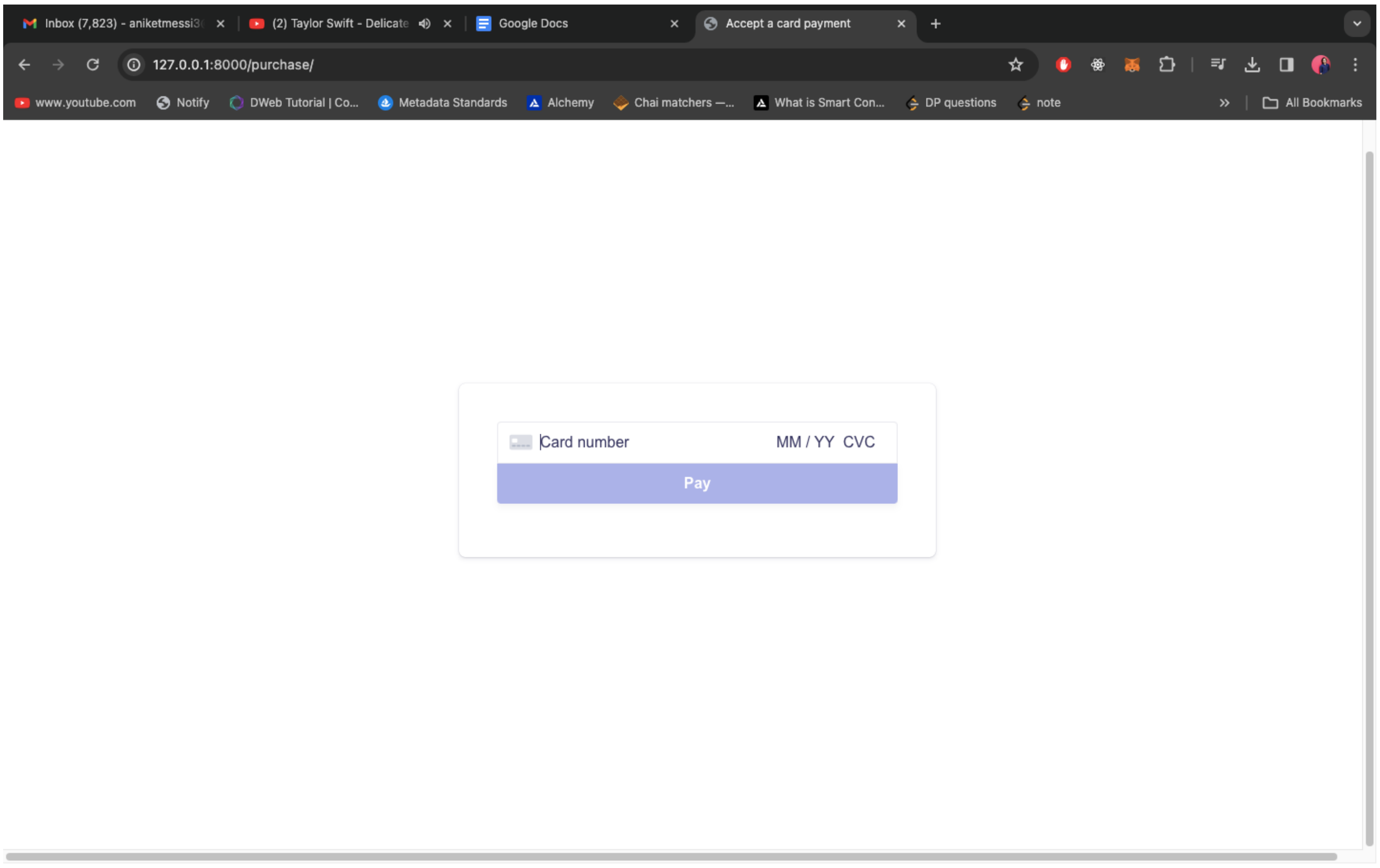


Timer:0:53 seconds

Congratulation on finishing this problem!

```
b'----------------------------------------------------------------------
Ran 7 tests in 0.001s

OK
'
```



Inbox (7,823) - aniketmessi3
(2) Taylor Swift - Delicate
Google Docs
Accept a card payment
127.0.0.1:8000/purchase/
www.youtube.com
Notify
DWeb Tutorial | Co...
Metadata Standards
Alchemy
Chai matchers —...
What is Smart Con...
DP questions
note
All Bookmarks
Card number
MM / YY
CVC
Pay

3. Testing

The testing phase of the Upalgo project plays a pivotal role in ensuring the reliability, functionality, and security of the platform. Rigorous testing was conducted to validate different aspects of the application.

3.1. Unit Testing

Individual components of Upalgo underwent unit testing to verify their functionality in isolation. This included testing backend API endpoints, frontend React components, and database interactions. Unit tests were instrumental in identifying and addressing issues at an early stage of development.

3.2. Integration Testing

Integration testing focused on evaluating the seamless interaction between different modules of the application. The integration of GitHub OAuth, Stripe payment processing, and the overall coordination between frontend and backend components were thoroughly tested. This phase aimed to identify and resolve any potential compatibility issues.

3.3. Performance Testing

Performance testing evaluated the responsiveness and efficiency of Upalgo under different scenarios. This included simulating various user loads and assessing the platform's stability.

Performance testing ensured that Upalgo could handle concurrent user interactions without compromising speed or functionality.

3.4. Security Testing

Security testing was a critical aspect of the Upalgo testing phase. GitHub OAuth and Stripe payment integration were scrutinized to ensure the secure handling of user data and financial transactions. Vulnerability assessments and penetration testing were conducted to identify and address potential security loopholes.

4. Conclusion: Upalgo Project

The completion of the Upalgo project marks a significant milestone in the journey to create a dynamic and user-centric coding challenge platform. The combination of thoughtful design, meticulous development, and comprehensive testing has culminated in a platform poised to make a positive impact in the coding community.

4.1. Achievements

- Successfully implemented a diverse set of 10 algorithmic questions catering to both unpaid and paid user segments.
- Seamless integration of GitHub OAuth for secure and user-friendly authentication.
- Implementation of Stripe payment processing for a sustainable monetization model.
- Robust technical architecture using Django, React.js, and MySQL, providing a scalable and efficient platform.
- Thorough testing, ensuring reliability, performance, and security across the entire application.

4.2. Lessons Learned

The development of Upalgo provided valuable insights into the complexities of building a coding challenge platform. From addressing payment security concerns to balancing question difficulty, each challenge presented an opportunity for learning and improvement. User feedback during testing phases was particularly instrumental in shaping the final product.

4.3. Future Directions

As Upalgo prepares for its launch, the future holds exciting possibilities. The integration of user analytics, community features, and continuous addition of high-quality questions are on the horizon. The project team remains committed to evolving Upalgo based on user needs and technological advancements in the coding landscape.

In conclusion, Upalgo is not just a project; it's a testament to innovation, collaboration, and the pursuit of excellence in the realm of coding challenges. As users embark on their coding journeys with Upalgo, the platform stands ready to be a catalyst for skill development, community engagement, and the joy of solving algorithmic puzzles.

Bibliography

1. Johnson, M. (2022). "Mastering Django: Web Development Simplified." O'Reilly Media.

2.React.js Documentation. (2022). Retrieved from https://reactjs.org/docs/getting-started.html

3. MySQL Documentation. (2022). Retrieved from https://dev.mysql.com/doc/

4.GitHub documentation- https://docs.github.com/en/developers/apps/building-oauth-apps

5. Stripe API Documentation. (2022). Retrieved from https://stripe.com/docs

6. LeetCode. (2022) - https://leetcode.com/

7. Upalgo GitHub Repository. (2022). Retrieved from https://github.com/aniik8/Up-Algo