CHAPTER 1: Introduction

1.1 Aim of the Project

Cricket streaming rights sky-rockets every new season, but still there is very limited innovation here. The task of classifying and generating highlights is currently done manually, but could be automated to some extent to make the process more dynamic. With this project we plan to build a Ball by Ball Event Classification and Highlights generation from a cricket match using deep learning approach.

1.2 Project Scope

Our website caters to a diverse audience of cricket enthusiasts, including casual viewers seeking updated information and personalized highlights, sport analysts and commentators seeking deeper insights for informed commentary, and media and content creators, including social media influencers and journalists, looking for engaging content and discussion points to share with their followers.

1.3 Project Objective

We aim to label each moment of the match, making it easy to display events and revisit them later if missed in real-time. Additionally, we plan to optimize the system architecture to provide low-latency event classification. The highlights will be dynamically generated based on the identified key events, with consideration of the user's preferences.

1.4 Project Modules

1.4.1. User registration and Authentication: This module is the entry point to a customized experience on our platform. Users can register and set up their profiles, which guarantees a safe login process. This module not only secures user information but also facilitates a personalized experience by enabling users to specify their preferences, keep track of their activity.

1.4.2. Event classification and display: At the heart of our platform, this module utilizes deep learning algorithms to classify live cricket events such as wickets, fours, sixes, and none. The classified events are displayed in real-time, offering users an insightful breakdown of the ongoing match. This feature enriches the viewing experience helps catching up important events.

1.4.3. Customized highlights generation: Catering to diverse user preferences, this module allows users to generate personalized match highlights. Whether it's a compilation of every boundary in a match or a focus on key wickets, users can create their own highlight reels. This not only enhances engagement but also allows fans to relive their favorite moments from the match as per their interests.

1.4.4. Prediction Leader-board: Engaging users in a fun and competitive way, the Prediction Leader-board module lets users predict outcomes of live match events. The predicted answers are compared with classified event by our algorithm. Correct predictions earn points, placing users on a leader-board where they compete with fellow cricket enthusiasts. This module adds an interactive layer to match viewing, and creates opportunities for sponsors for their giveaways.

1.5 Project basic requirements

1.5.1. Hardware Requirements:

5.1.1. A system with a minimum of 8GB RAM.

5.1.2. A Multi-core processor, intel® i-7 4-core.

5.1.3. A Intel® UHD Graphics 620, for training classification algorithm video processing.

1.5.2. Software Requirements:

5.2.1. React.js framework for front-end development.

5.2.2. MongoDB as a database management system.

5.2.3. Express.js framework for back-end development.

5.2.4. Git for version control.

5.2.5. TensorFlow, PyTorch and Keras Frameworks for building networks.

5.2.6. NumPy, Pandas Data Processing Libraries for handling datasets.

CHAPTER 2: Analysis, Design Methodology, and Implementation Strategy

2.1 Comparison of Existing Applications with Your Project

2.1.1 From Subjectivity to Precision:

Traditional manual methods rely on human interpretation, which can introduce variability in event classification. Our automated system, on the other hand, uses deep learning to ensure objective and consistent accuracy, thereby enhancing the quality of event recognition.

2.1.2. Speed of Delivery:

The manual creation of highlights is a slow and labor-intensive process. Conversely, our automated approach promises near-instant highlight generation, transforming hours of footage into captivating moments in real-time.

2.1.3. Scaling New Heights:

Manual processing struggles to keep pace with multiple simultaneous matches, as it is limited by human capacity. Our system breaks these boundaries, effortlessly scaling to accommodate numerous games.

2.1.4. Customized Viewing Experience:

Whereas manual methods offer a one-size-fits-all approach, our system stands out by personalizing content. It uses viewer preference data to tailor highlights to individual tastes, thereby deepening viewer engagement.

2.1.5. Economic Efficiency:

The shift from manual labor to an automated system translates into significant cost savings. While the initial investment focuses on technology development, the long-term efficiency and reduced reliance on manual processes make it economically advantageous.

2.1.5. Uniform Quality Assurance:

Human-dependent processes inherently vary in consistency. Our automated system, immune to such fluctuations, promises a reliable and uniform quality of output, ensuring every highlight is of top-notch quality.

2.2 Project Feasibility Study

2.2.1. Technical Feasibility:

This involves ability to implement and fine-tune deep learning models for accurate cricket event classification through research and development. Additionally, the handling of high-quality cricket match data for model training and validation need to be evaluated. The availability of necessary hardware like GPUs for deep learning and software like programming languages and frameworks also needs to be considered.

2.2.2. Legal Feasibility:

This includes understanding and complying with legal restrictions related to broadcasting and replaying cricket match footage. Adhering to applicable data protection laws in handling user data, especially if personalization features are incorporated is also necessary. Preparing for legal agreements necessary for content access and partnerships is also essential.

2.2.3. Market Feasibility:

Market demand analysis needs to be conducted to investigate the demand for AI-driven sports event classification and highlight generation among cricket audiences and media outlets. Identifying existing market solutions and positioning our project in terms of innovation and value addition is also important. Exploring collaboration possibilities with sports media entities,

streaming platforms, and cricket boards is vital. Considering various revenue generation strategies including, subscriptions, advertising, and licensing is also considered.

2.2.4. Operational Feasibility:

This includes designing a user interface that is intuitive for both general users seeking match highlights and administrators overseeing system operations. It also involves ensuring that the system is reliable, especially during peak times like live matches, and establishing a robust maintenance plan. Planning for potential scaling needs to accommodate large user bases and high data volumes during major tournaments is also essential.

2.3 Project Timeline Chart

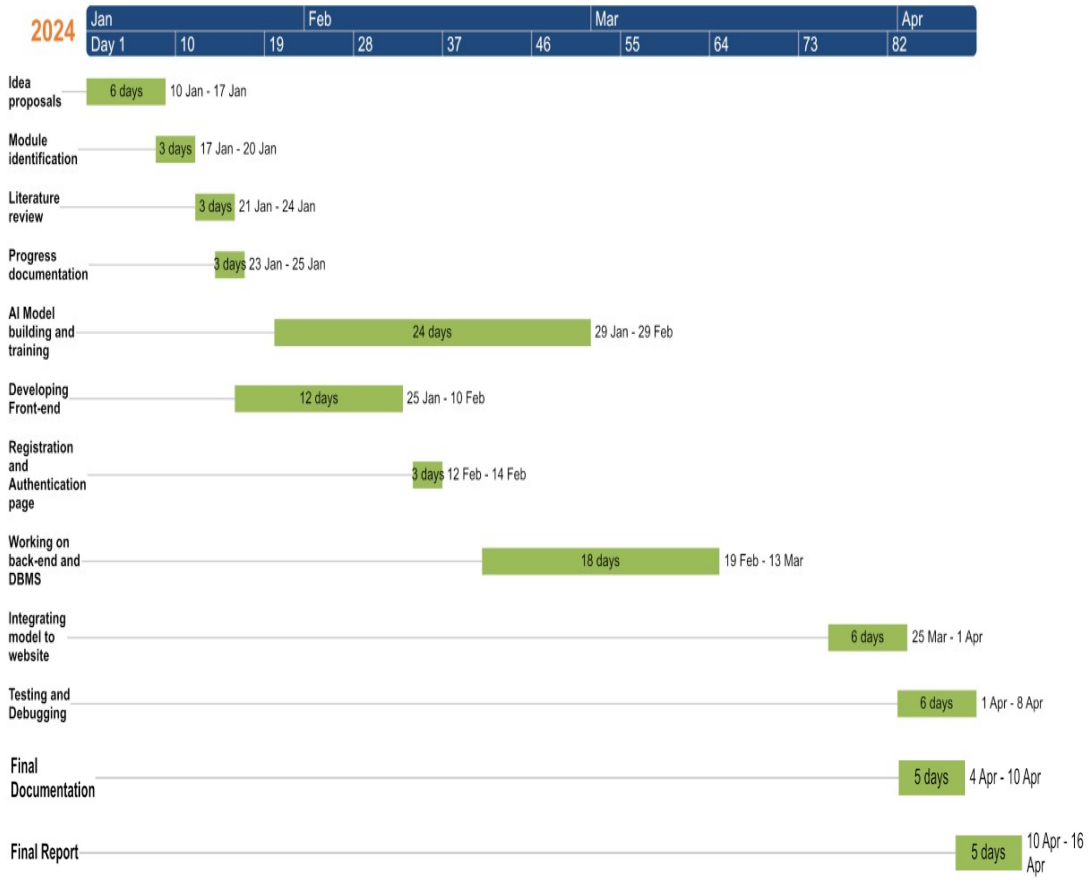


Figure 2.3: Time Chart

2.4 Detailed Modules Description

2.4.1. User registration and Authentication: This module is the entry point to a customized experience on our platform.

Functions:

- Users can register and set up their profiles, which guarantees a safe login process.
- This module not only secures user information but also facilitates a personalized experience by enabling users to specify their preferences, keep track of their activity.

2.4.2. Event classification and display: At the heart of our platform, this module utilizes deep learning algorithms to classify live cricket events such as wickets, fours, sixes, and none.

Functions:

- The classified events are displayed in real-time, offering users an insightful breakdown of the ongoing match.
- This feature enriches the viewing experience helps catching up important events.

2.4.3. Customized highlights generation: This module Caters to diverse user preferences, this module allows users to generate personalized match highlights.

Functions:

- Whether it's a compilation of every boundary in a match or a focus on key wickets, users can create their own highlight reels. This not only enhances engagement but also allows fans to relive their favorite moments from the match as per their interests.

2.4.4. Prediction Leader-board: Engaging users in a fun and competitive way, the Prediction Leader-board module lets users predict outcomes of live match events.

Functions:

- The predicted answers are compared with classified event by our algorithm. Correct predictions earn points, placing users on a leader-board where they compete with fellow cricket enthusiasts.
- This module adds an interactive layer to match viewing, and creates opportunities for sponsors for their giveaways.

2.5 Project SRS

2.5.1 Use Case Diagrams

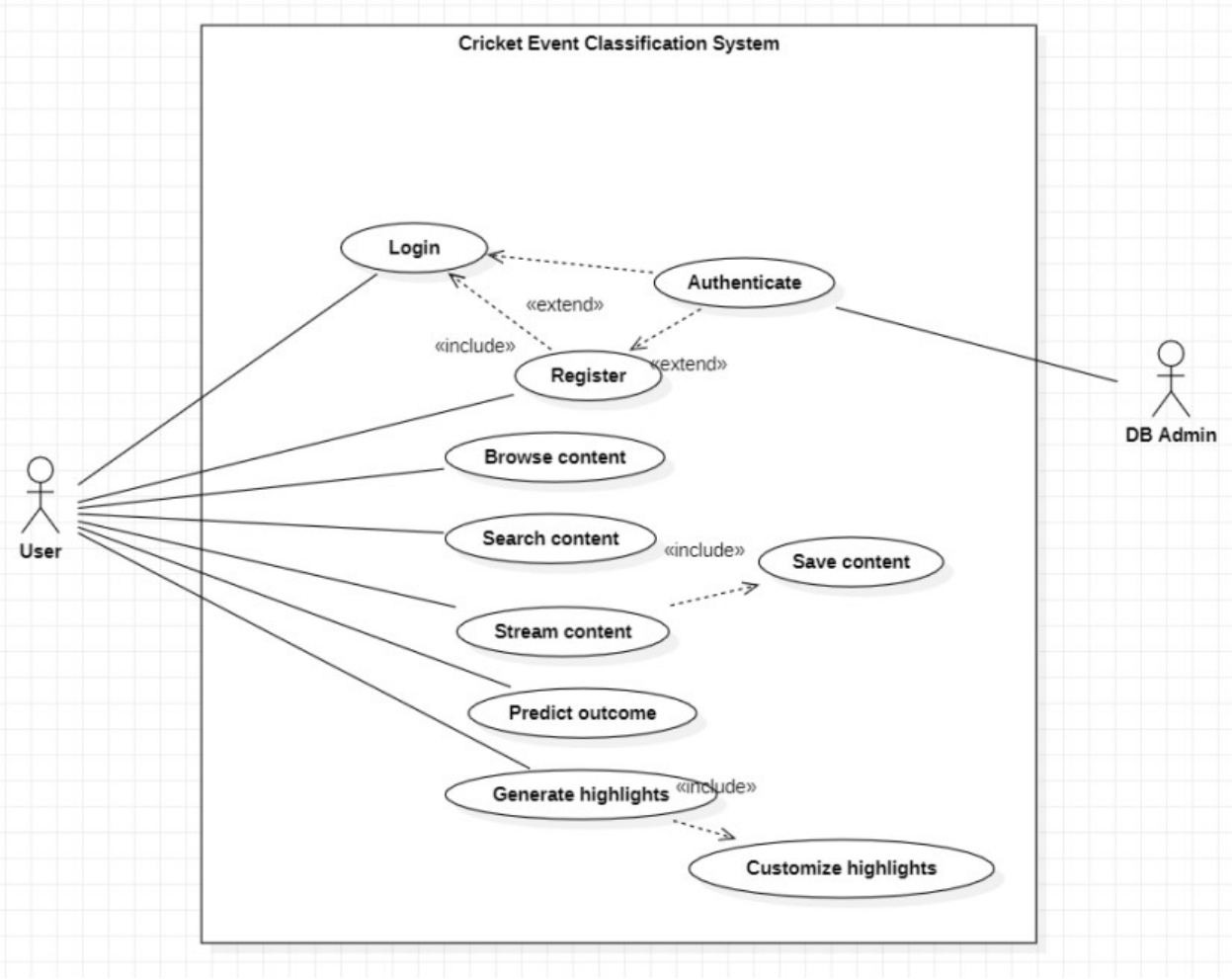


Figure 2.5.1: Use case Diagram

2.5.2 Data Flow Diagram:

Level-0:

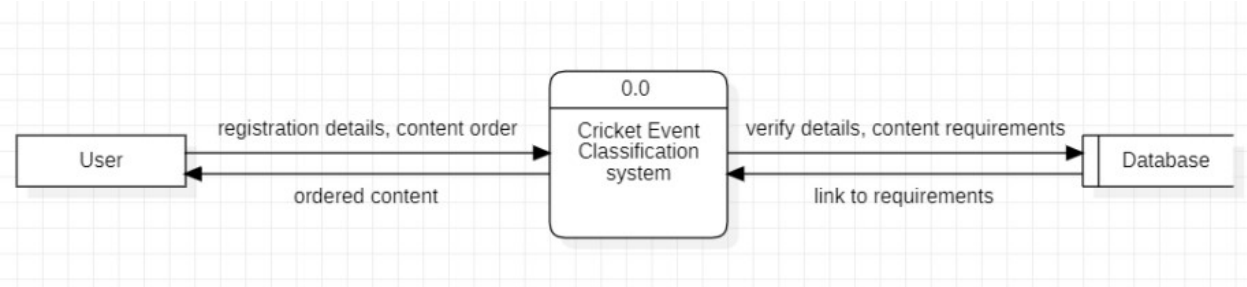


Figure 2.5.2: DFD level 0

Level-1:

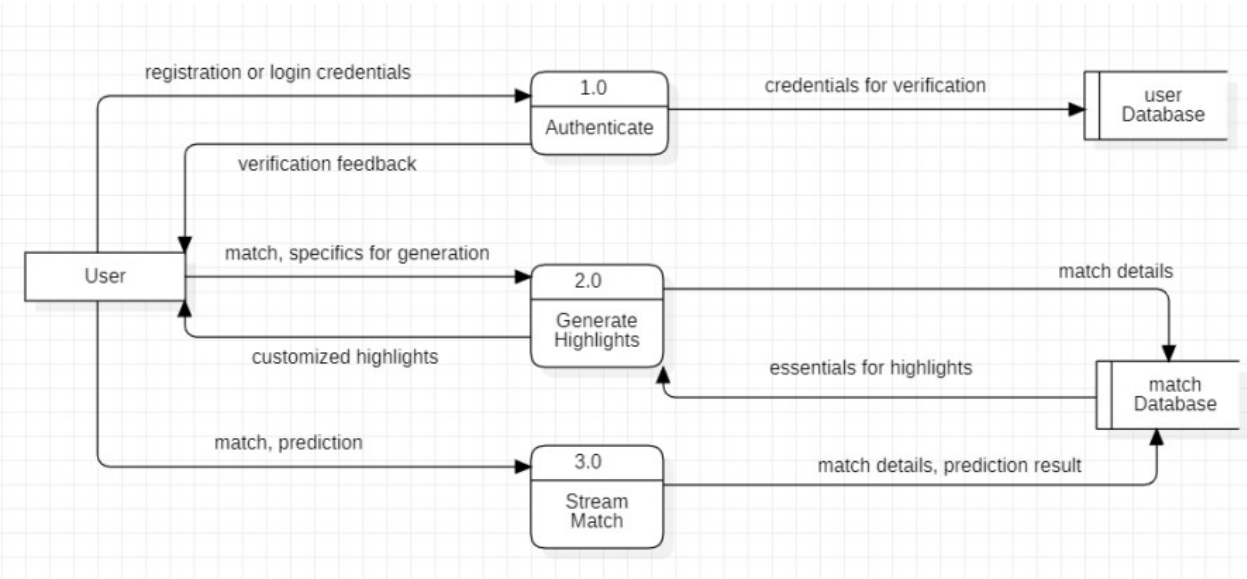


Figure 2.5.2: DFD level 1

Level-2:

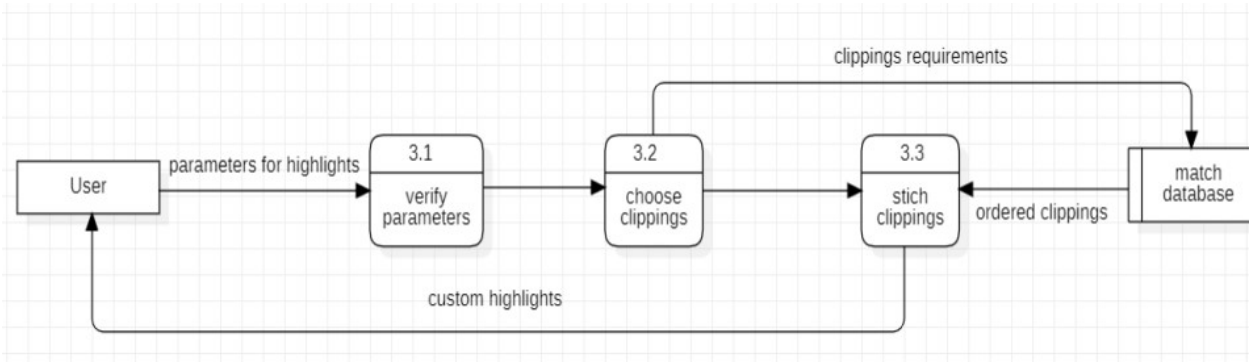


Figure 2.5.2: DFD level 2

2.5.3 Entity Relationship Diagrams

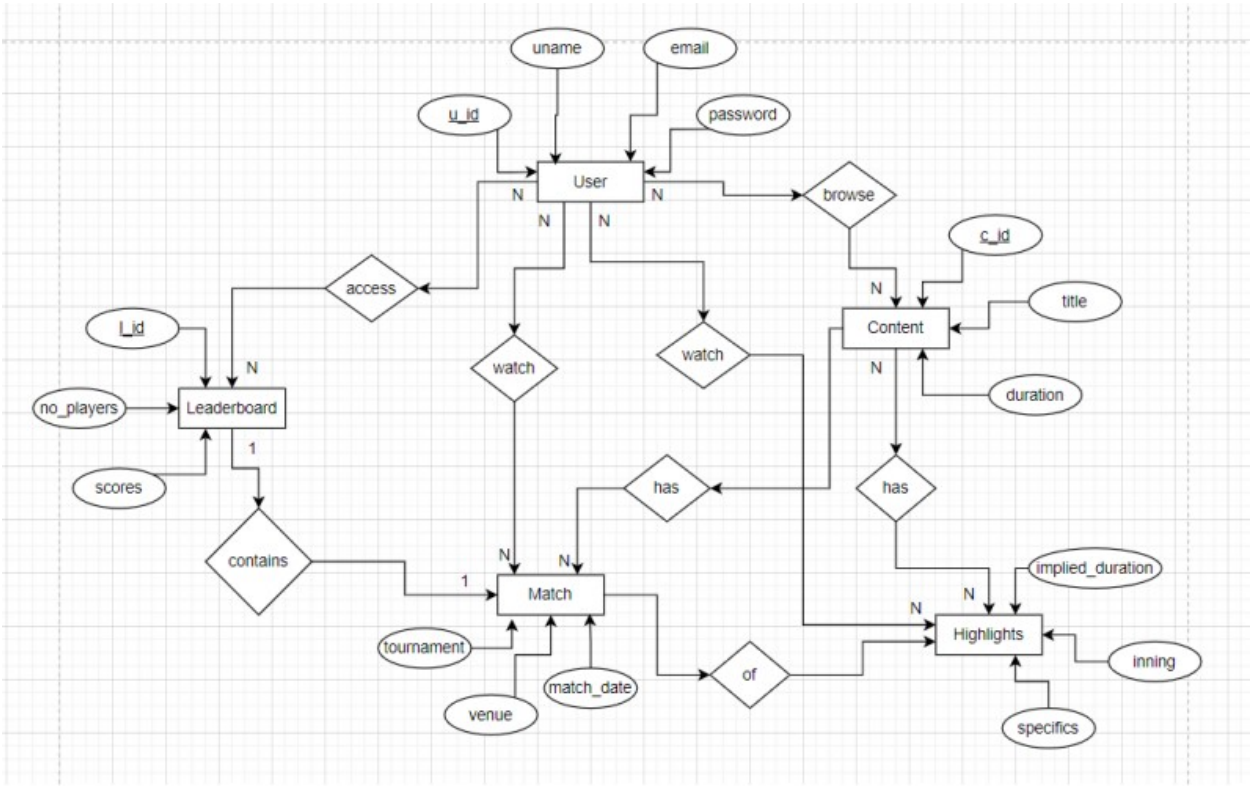


Figure 2.5.3: E-R Diagram

2.5.4 Event Trace Diagram

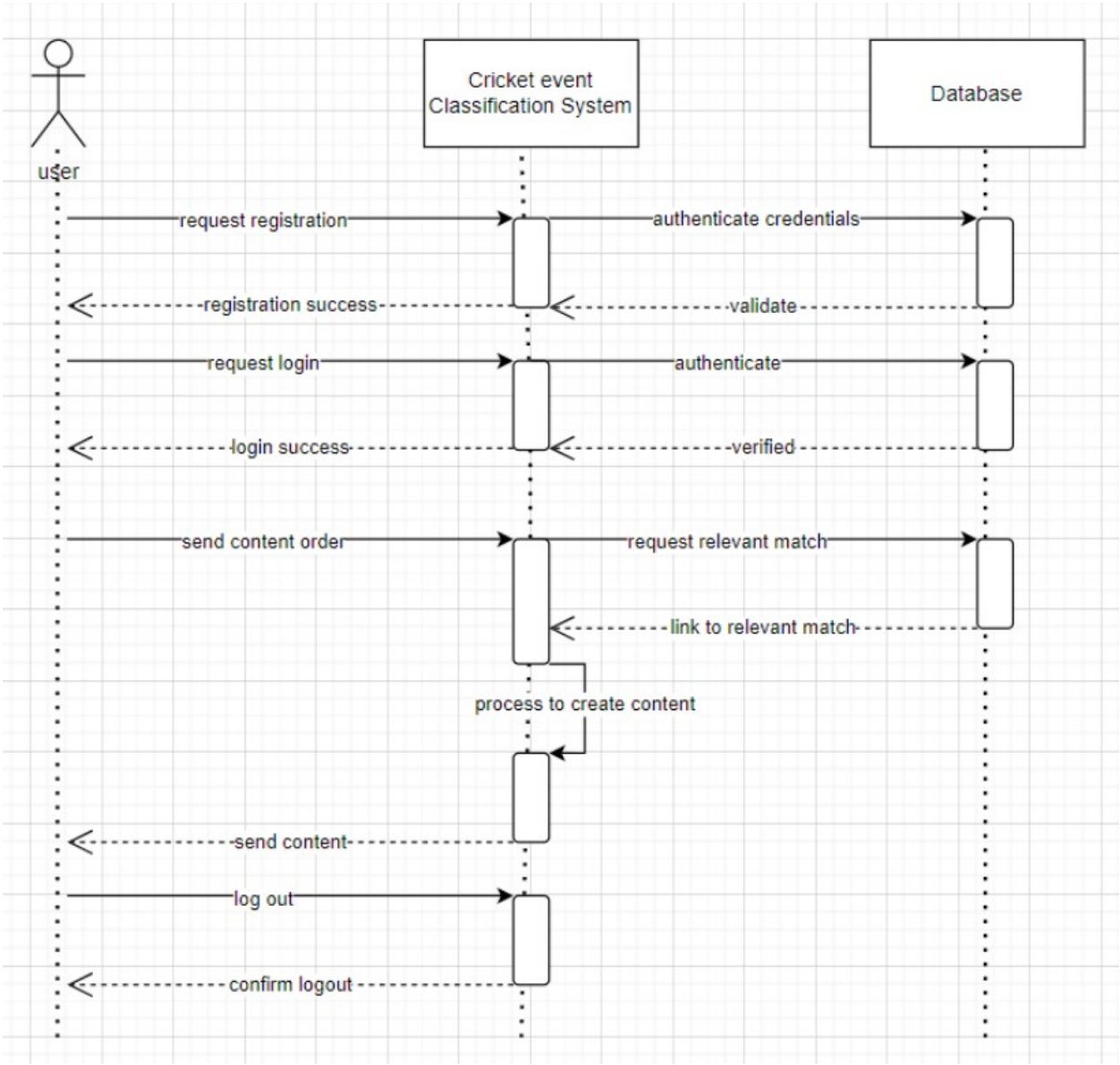


Figure 2.5.4: Event Trace Diagram

2.5.5 State diagram

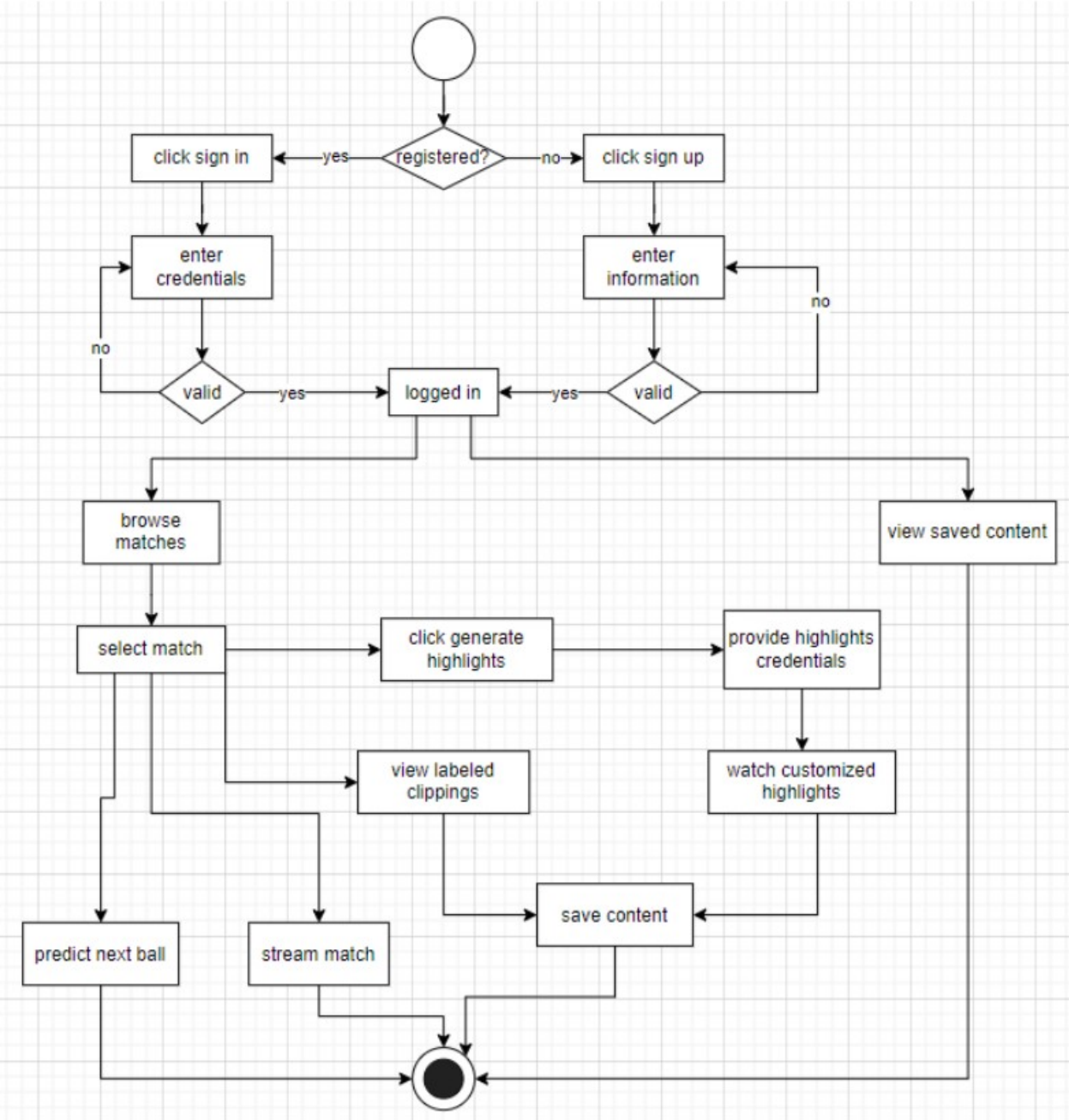


Figure 2.5.5: State Diagram

2.5.6 Class diagram

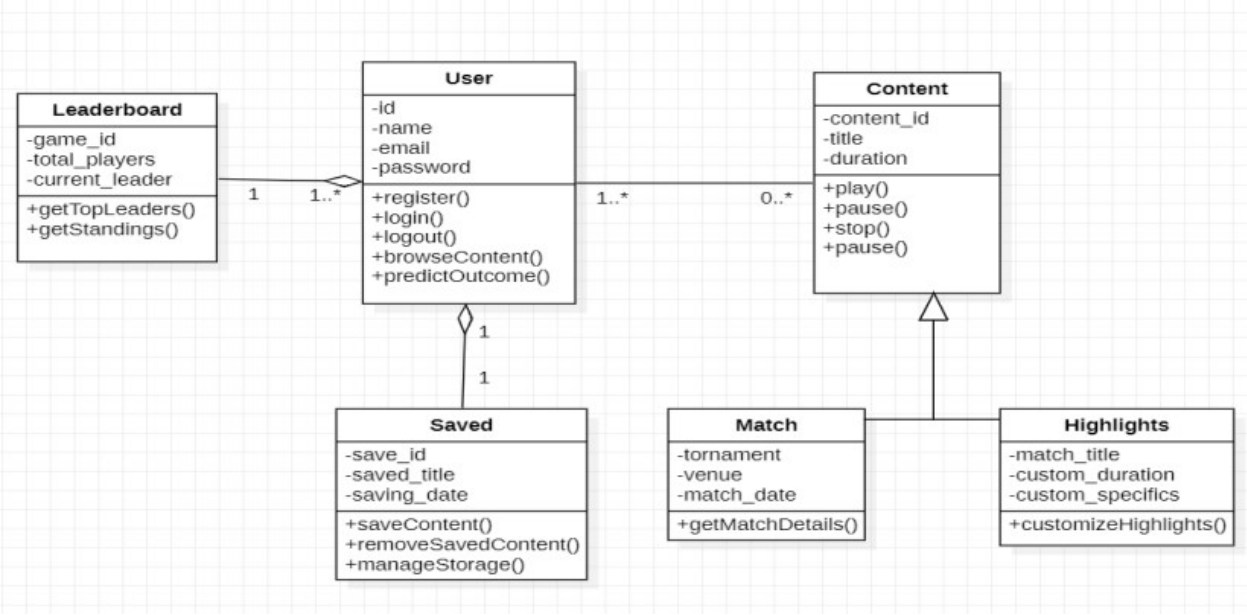


Figure 2.5.6: Class Diagram

2.6 Database Design and Normalization

2.6.1 Table Name: Match

Description: to store match details

Primary Key: Match_id

Sr. No.	Name	Data Type	Constraint	Description
1	Match_id	INT	Primary Key	Unique match id
2	Match_tournament	VARCHAR (50)	Not Null	Store the name of tournament
3	Match_name	VARCHAR (100)	Not Null	Match's unique name
4	Match_date	DATE	Not Null	Match's date
5	Match_venue	VARCHAR (100)	Not Null	Venue of the match

2.6.2 Table Name: Classified event

Description: To store information of classified event

Primary Key: Clip_id

Sr. No.	Name	Data Type	Constraint	Description
1	Match_id	INT	Foreign Key	Unique match id
2	Clip_id	INT	Primary Key	Unique id of every clip
3	Start_time	INT	-	Starting time of clip
4	End_time	INT	-	Ending time of clip
5	Event	VARCHAR	-	Name of classified event

2.6.3 Table Name: User

Description: to store user credentials

Primary Key: Email_id

Sr. No.	Name	Data Type	Constraint	Description
1	User_name	VARCHAR (50)	Unique	Unique identifier for the user
2	Email_id	VARCHAR (50)	Primary_key	Email address of the user
3	Password	VARCHAR	Not Null	User's Password

2.6.4 Table Name: Leader Board

Description: to store information regarding leader board winnings

Foreign Key: Match_id

Sr. No.	Name	Data Type	Constraint	Description
1	Match_id	INT	Foreign Key	Unique match id
2	User_name	VARCHAR(100)	Not Null	Unique user identity
3	Points	INT	-	Points user scored

CHAPTER 3: Implementation and Testing

3.1 Software and tools

3.1.1 Front-End Development

For the frontend development of the cricket event classification and highlights generation platform, React.js is utilized as the primary framework due to its component-based architecture and efficient rendering. Alongside React.js, HTML, CSS, and JavaScript (ES6+) were employed for structuring web pages, styling the user interface, and adding interactivity. Additionally, Redux is used for state management, and Bootstrap streamlined development with pre-designed components and responsive layouts.

3.1.2 Back-End Development

Express.js served as the framework for building RESTful APIs and handling server-side logic, while MongoDB is used as the database management system for storing and retrieving data related to user profiles, match events, and preferences. Node.js provided the runtime environment for server-side JavaScript execution, and Git for version control. FFMpeg was employed for merging video files for highlights generation. For the requirement of continuously sending data to client from server for Server-Sent Events(SSE).

3.2 User Interface and Snapshot

3.2.1 Login

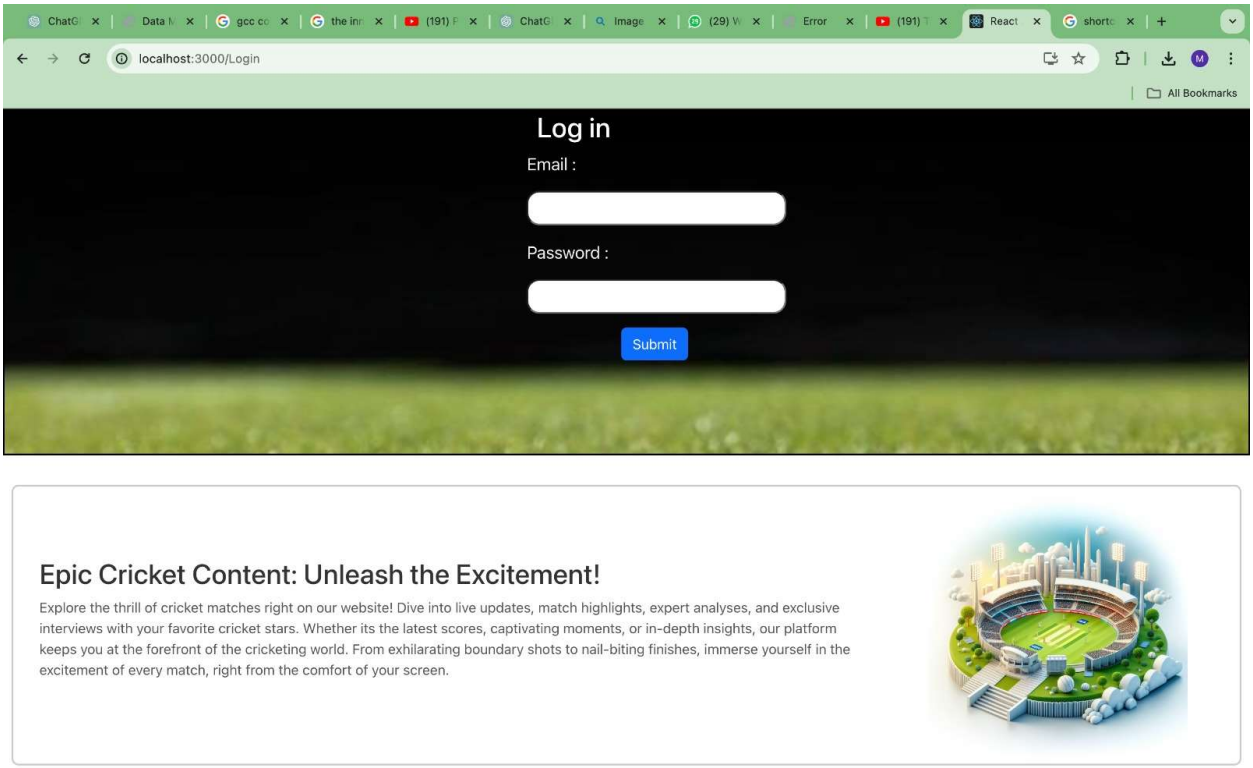


Figure 3.2.1: login system

The login page features essential fields for users to input their email address and password, along with a prominent "Submit" button. Once users enter their credentials and click "Submit," the data is sent to the Mongoose Compass, where it is securely stored in the database. This process ensures that user login information is safely managed and accessible for authentication purposes, contributing to a seamless and secure user experience on the platform.

3.2.2 Sign-up

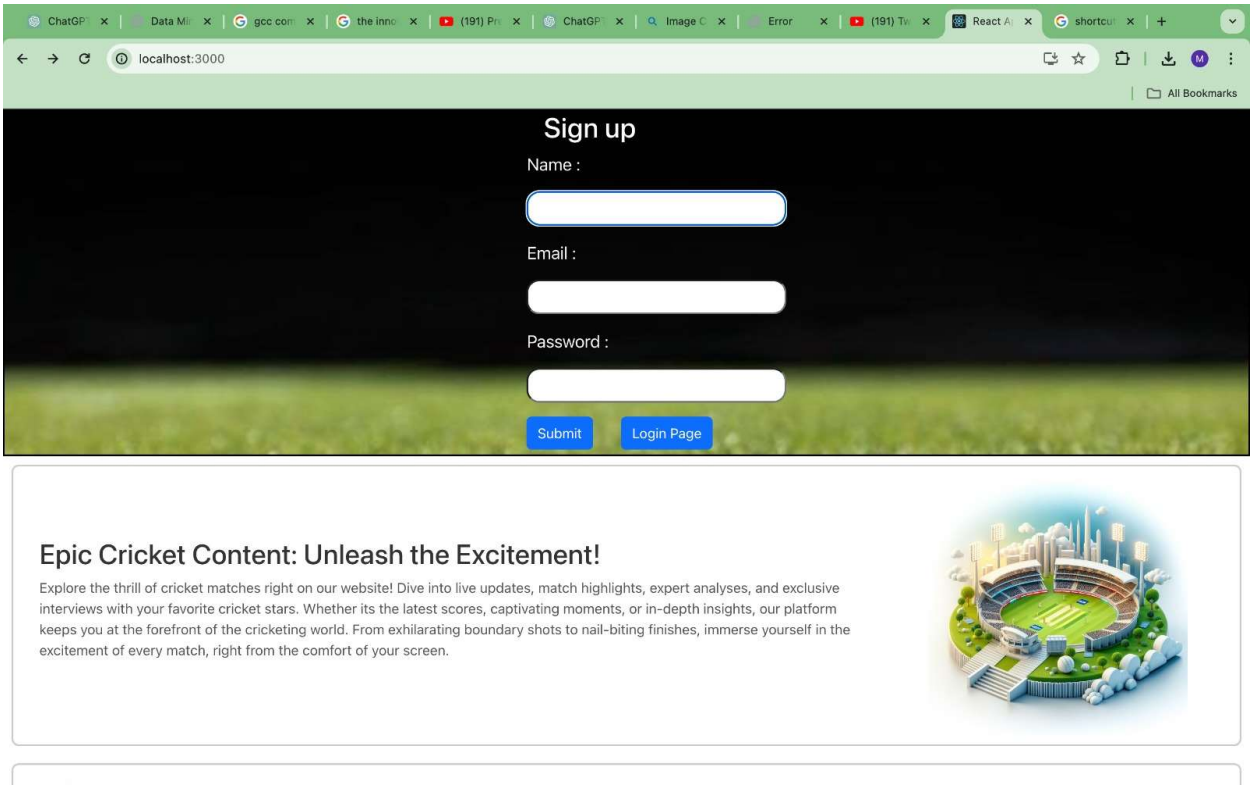


Figure 3.2.2: sign-up

The signup page provides users with fields to input their email address, password, and username, essential for creating a new account. Upon completion, users are redirected 16 to the home page, where they can begin exploring the platform's features and content. This streamlined process ensures a smooth transition for new users, allowing them immediate access to the platform's offerings after registration.

3.2.3 Home-page

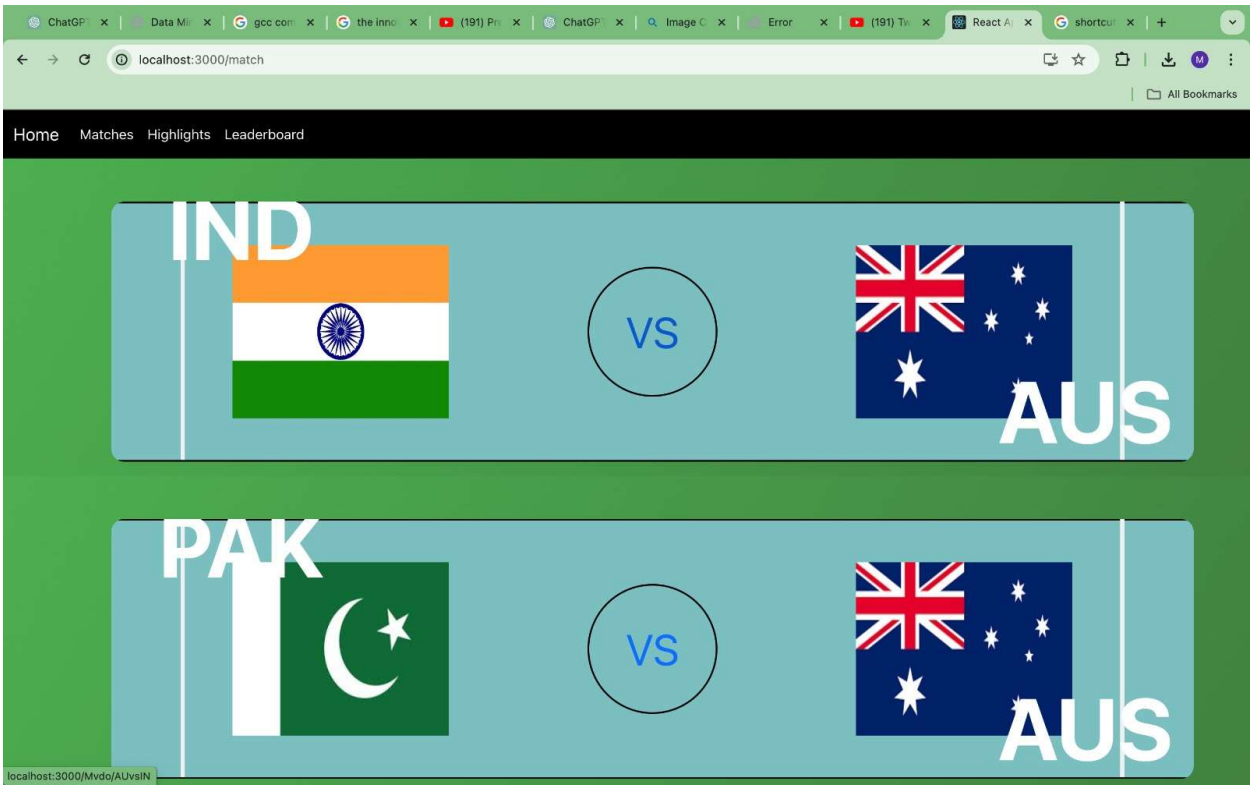


Figure 3.2.3: Home page

Step into the world of cricket with our website's inviting homepage, where a sleek navigation bar awaits to guide you seamlessly through our platform's offerings. Scroll down to discover captivating visuals and a concise description of our features. At the bottom, our footer houses all contact details, ensuring easy communication. Welcome to an immersive cricket experience designed just for you.

3.2.4 Event Classification and Display:

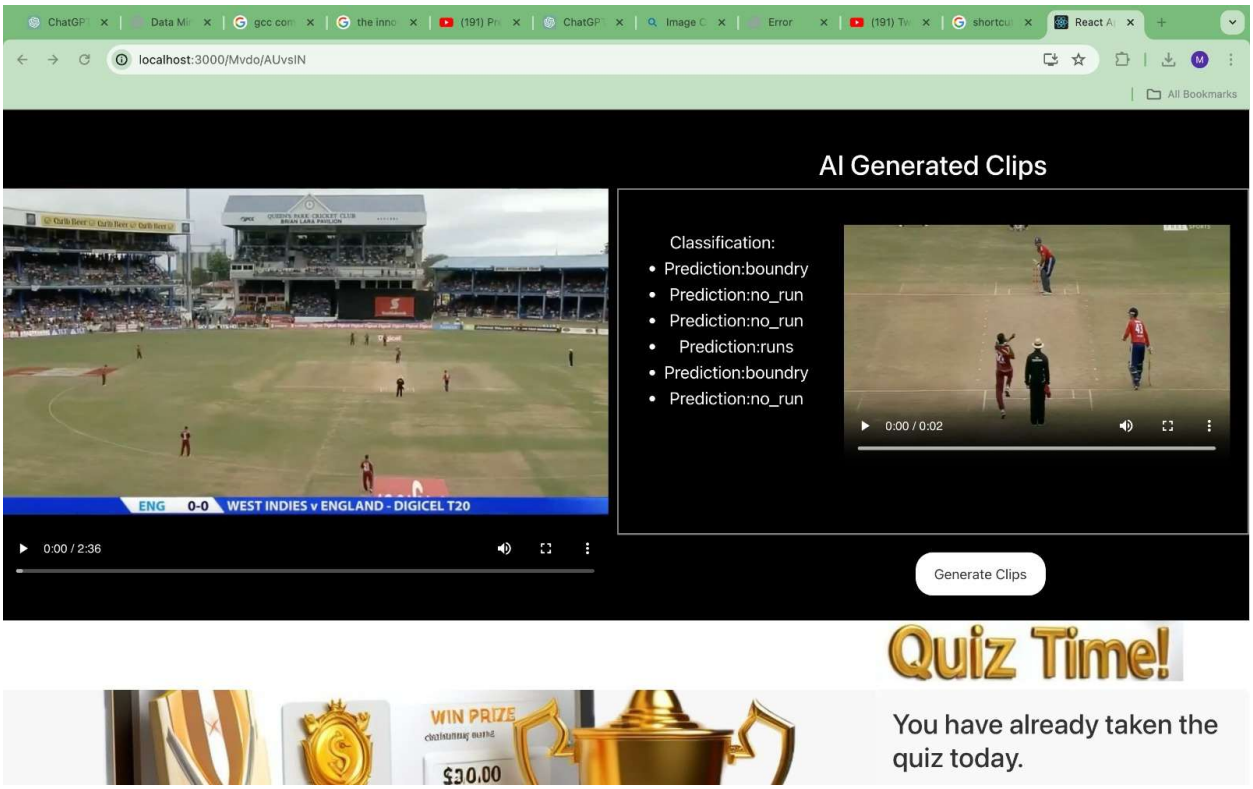


Figure 3.2.4.1: Event Classification and Display

The "Matches" section serves as a hub for users to access ongoing matches, displaying a list of current matches available for viewing. Upon selection of a specific match, users are redirected to the live match page where event classification is predicted in real-time.

Alongside the match classification, a side box displays the Prediction Leaderboard, engaging users in interactive match predictions. This dynamic feature enriches the viewing experience by providing live match updates and fostering user participation through predictions, enhancing overall engagement with the platform.

The classification model employed in the platform analyzes cricket videos, categorizing each ball delivery into four distinct classes: "No Run," "Boundary," "Wicket," and "Run." This model operates by leveraging deep learning algorithms to identify and classify the outcomes of each ball delivery in real-time. By accurately distinguishing between different event types, including

boundaries, wickets, and runs, the model enhances the platform's ability to provide users with comprehensive and informative match highlights, thereby enriching their viewing experience.

3.2.5 Customize highlights Generation

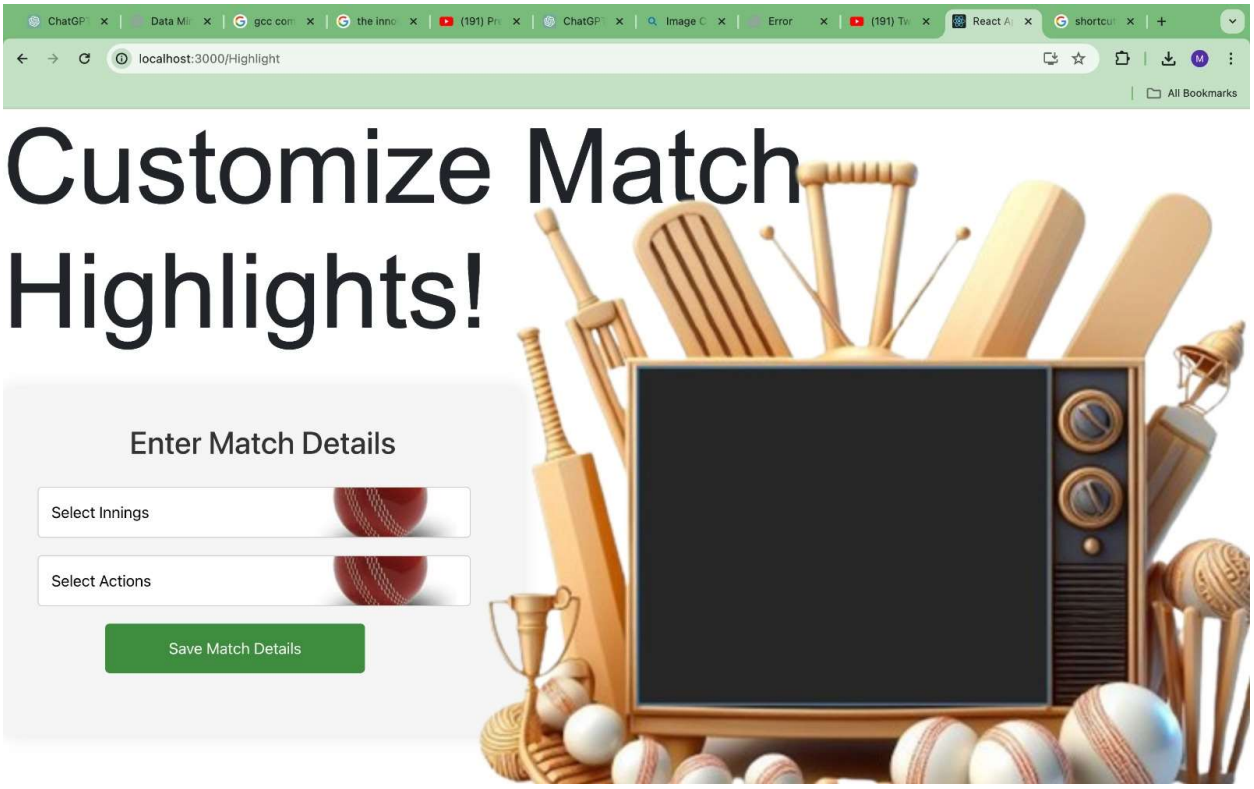
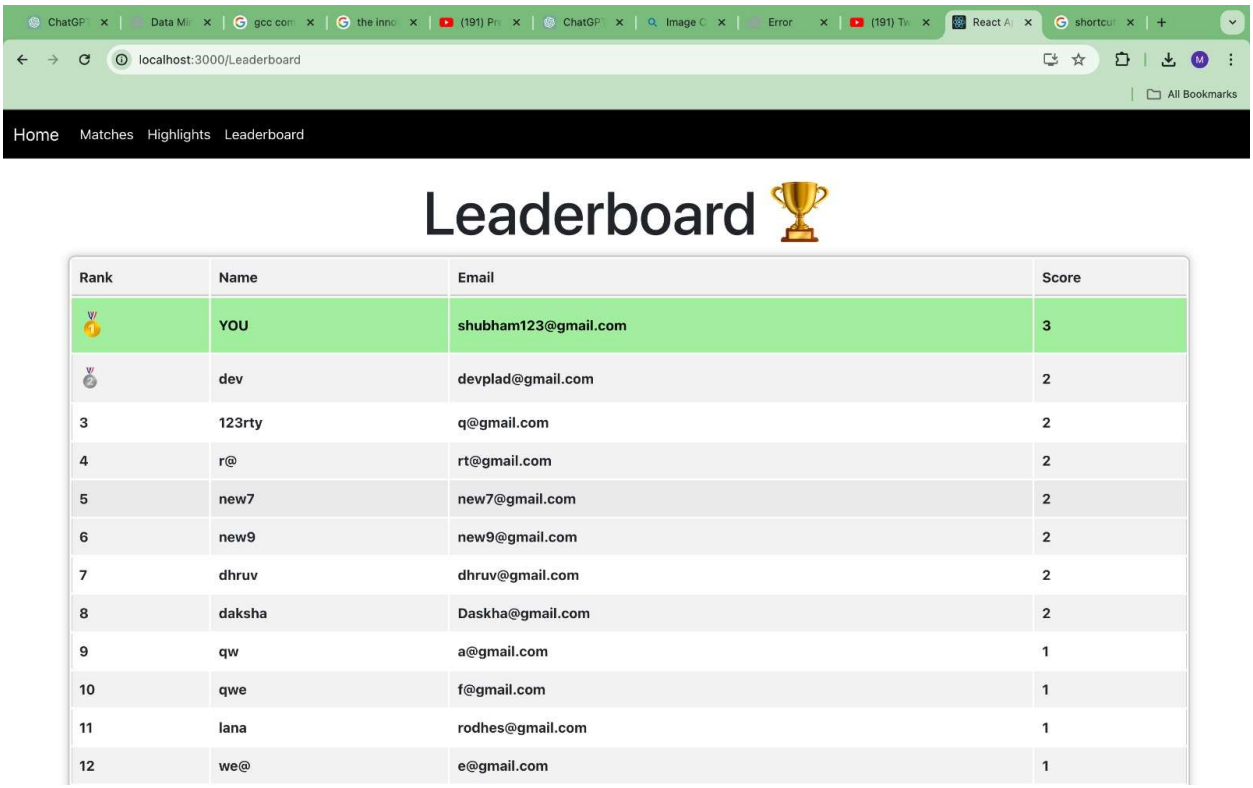


Figure 3.2.5: Customized Highlights

The Customized Highlights Generation feature empowers users to curate personalized match highlights tailored to their preferences. Users input their desired criteria, such as specific events like boundaries or wickets, and the platform generates customized highlight videos accordingly.

The screenshot showcases the user interface where users can select their preferences, with the adjacent section displaying the dynamically generated customized highlight videos in real-time. This interactive functionality enhances user engagement by allowing them to relive their favorite moments from the match according to their interests, thereby fostering a deeper connection with the game.

3.2.6 Leader Board



Rank	Name	Email	Score
1	YOU	shubham123@gmail.com	3
2	dev	devplad@gmail.com	2
3	123rty	q@gmail.com	2
4	r@	rt@gmail.com	2
5	new7	new7@gmail.com	2
6	new9	new9@gmail.com	2
7	dhruv	dhruv@gmail.com	2
8	daksha	Daskha@gmail.com	2
9	qw	a@gmail.com	1
10	qwe	f@gmail.com	1
11	lana	rodhes@gmail.com	1
12	we@	e@gmail.com	1

Figure 3.2.6: Leaderboard

To foster user participation, encourage community interaction, and create opportunities for sponsors to engage with the audience through giveaways and promotions, this feature is included. Here users can take a quiz on the website while streaming match, revisiting events or in between generating personalized highlights.

The leaderboard displays the usernames of participants along with their current point totals, ranking them in descending order based on their scores. Users can track their progress and compare their standings with other participants, fostering a sense of competition and camaraderie among users.The leaderboard is updated dynamically using technologies such as Server-Sent Events (SSE) or WebSocket connections.

3.3 Testing Using Use Cases

3.3.1 User Authentication:

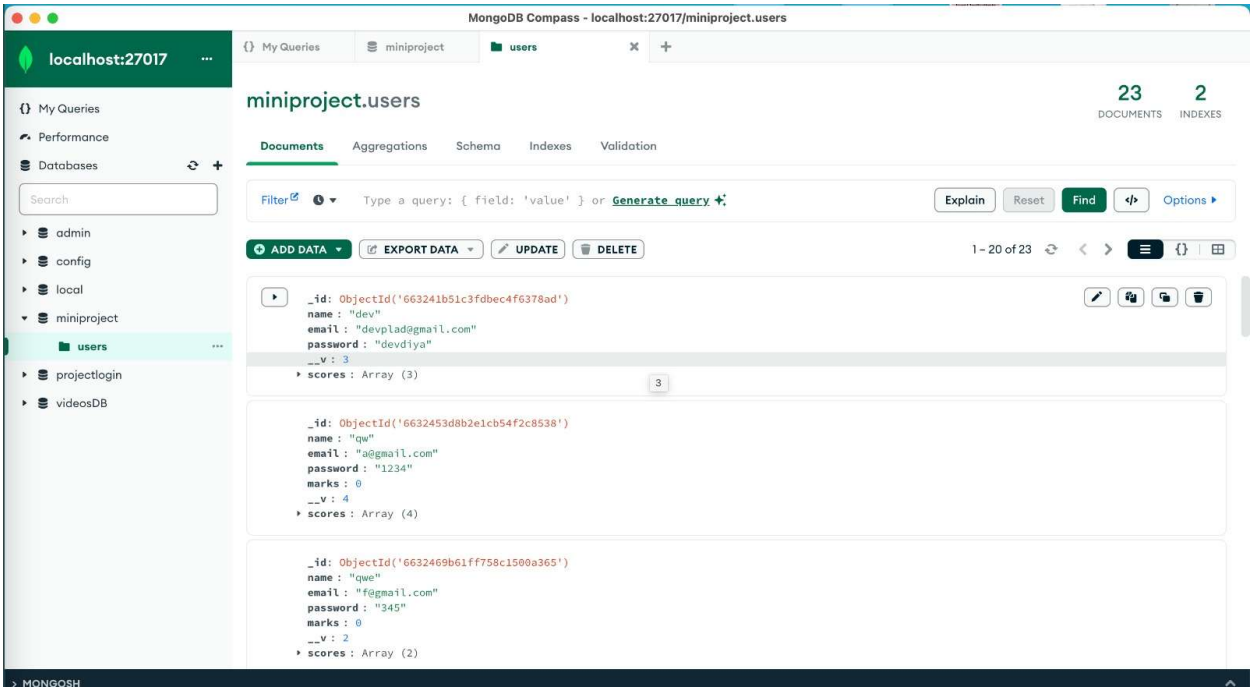


Figure 3.3.1: login-API

This test case focuses on the user authentication process, which includes sign-up and sign-in functionalities. The figure likely illustrates the flow of steps or the user interface elements involved in the authentication process, such as entering credentials, verifying information, and granting access to the application.

3.3.2 Clipping and Classification.

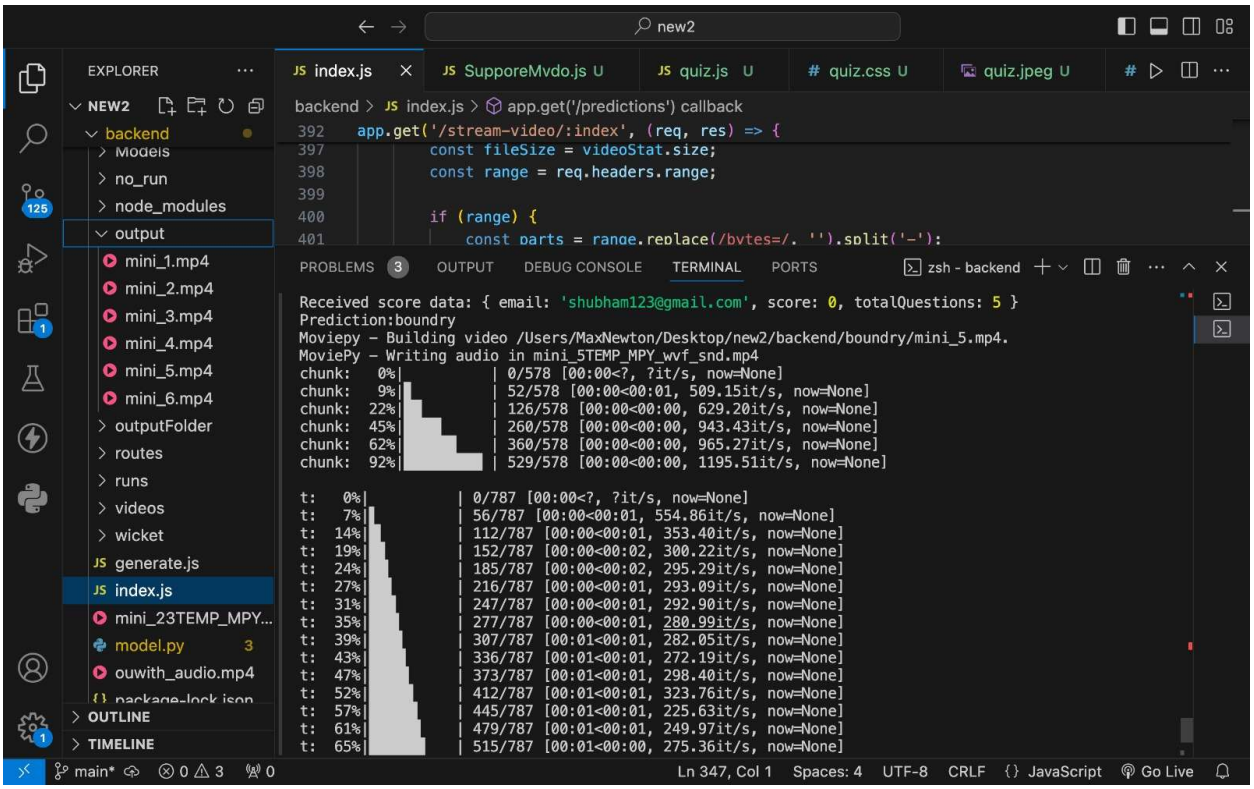


Figure 3.3.2: Clipping and Classification

This test case covers the backend process of the application, specifically the clipping and classification of data or content. The figure might depict the workflow or the architecture of the backend system, including components responsible for segmenting or extracting relevant clips from a larger dataset and classifying or categorizing those clips based on specific criteria or models.

3.3.3 Merging clips (For customized highlights).

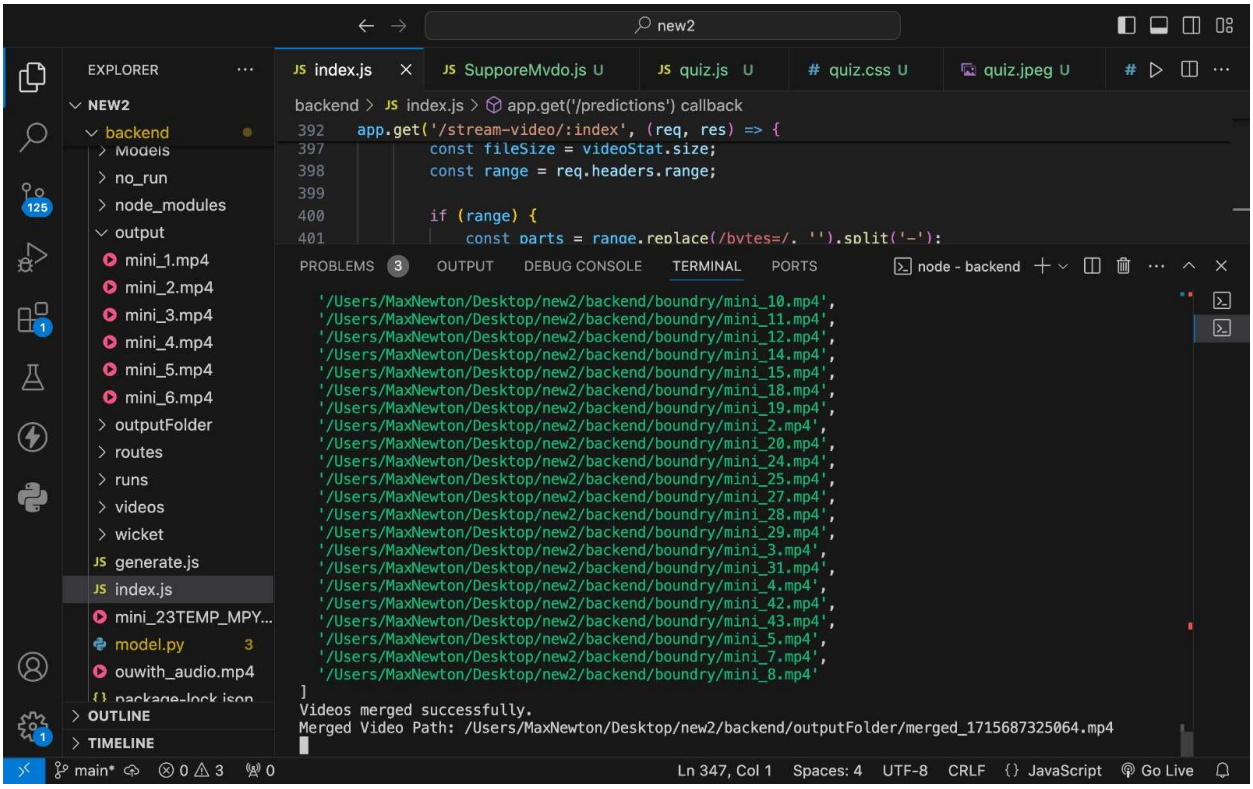


Figure 3.3.3: Merging clips

This test case addresses the generation of customized highlights by merging individual clips. The figure is likely illustrating the process or the algorithm involved in combining multiple clips, potentially based on user preferences or predefined rules, to create a curated or personalized highlight reel or summary.

CHAPTER 4: Conclusion and Future Work

4.1 Conclusion

Our application opened up new perspectives on cricket streaming. It also revealed AI's ability to transform many aspects of today's world for the betterment of humanity. Even though this project has challenges, we are confident that with additional work in this direction, we will be able to provide a more solid and production-ready solution. Nonetheless, our passion drove us to explore this subject, and we achieved our goals.

4.2 Future work

Owing to the complexity of this project, it lacks at some key parametes of accuracy and latency. In future we aim to put in some work to overcome this. some of the future works are as mentioned.

Because of our limitations, the dataset was comparatively smaller here because it was manually constructed. Therefore, using outsourcing to generate a large number of diverse dataset is one technique to improve accuracy. This may require us to slightly modify our model in order to significantly increase the accuracy of ball beginning and classification of event.

Implement optimizations in the system architecture and data processing pipeline to minimize latency in event classification and highlight generation. This could include optimizing algorithms for faster inference, utilizing hardware acceleration such as GPUs or specialized inference chips, and optimizing database queries and network communication for real-time performance.

Once it satisfies the required parameters which we like we hope to productionize this, may be on our own or under someones banner.

References

Websites:

Keras - https://keras.io/

Numpy - https://numpy.org/doc/

Cv2 - https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html

FFmpeg - https://pypi.org/project/ffmpeg-python/

Chokidar - https://www.npmjs.com/package/chokidar

Range - https://developer.mozilla.org/en-US/docs/Web/API/Range

MoviePy - https://pypi.org/project/moviepy/

Shutil - https://docs.python.org/3/library/shutil.html

Research papers:

Xingjian Shi, Zhourong Chen, Hao Wang ,Dit-Yan Yeung,Wai-Kin Wong, Wang-chun Woo. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting.

Kalpit Dixit, Anusha Balakrishnan. Deep Learning using CNNs for Ball-by-Ball Outcome Classification in Sports.

Kumail Abbas, Muhammad Saeed. Deep-Learning-Based Computer Vision Approach For The Segmentation Of Ball Deliveries And Tracking In Cricket.