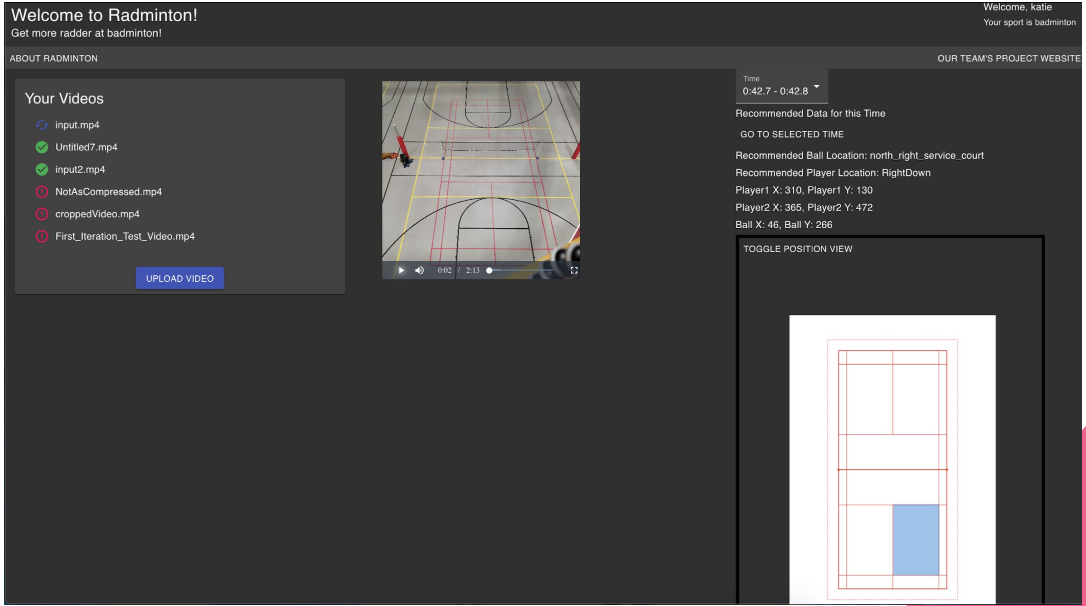
Radminton

Christion Barnes
Katelyn Sinn
John Rachid
Benjamin Kramer
Aiden McMinimy
Brian Guidarini

Advisor and Client:
Mitra Simanta- smitra@iastate.edu

High Level Overview

Radminton uses computer vision and modern web service technology to bring easy to use training tools to badminton players of all skill levels



Welcome to Radminton!

Get more radder at badminton!

Welcome, katie

Your sport is badminton

Your Videos

- input.mp4
- ✓ Untitled7.mp4
- ✓ input2.mp4
- ! NotAsCompressed.mp4
- ! croppedVideo.mp4
- ! First_Iteration_Test_Video.mp4

Time: 0:42.7 - 0:42.8

Recommended Data for this Time

Recommended Ball Location: north_right_service_court

Recommended Player Location: RightDown

Player1 X: 310, Player1 Y: 130

Player2 X: 365, Player2 Y: 472

Ball X: 46, Ball Y: 266

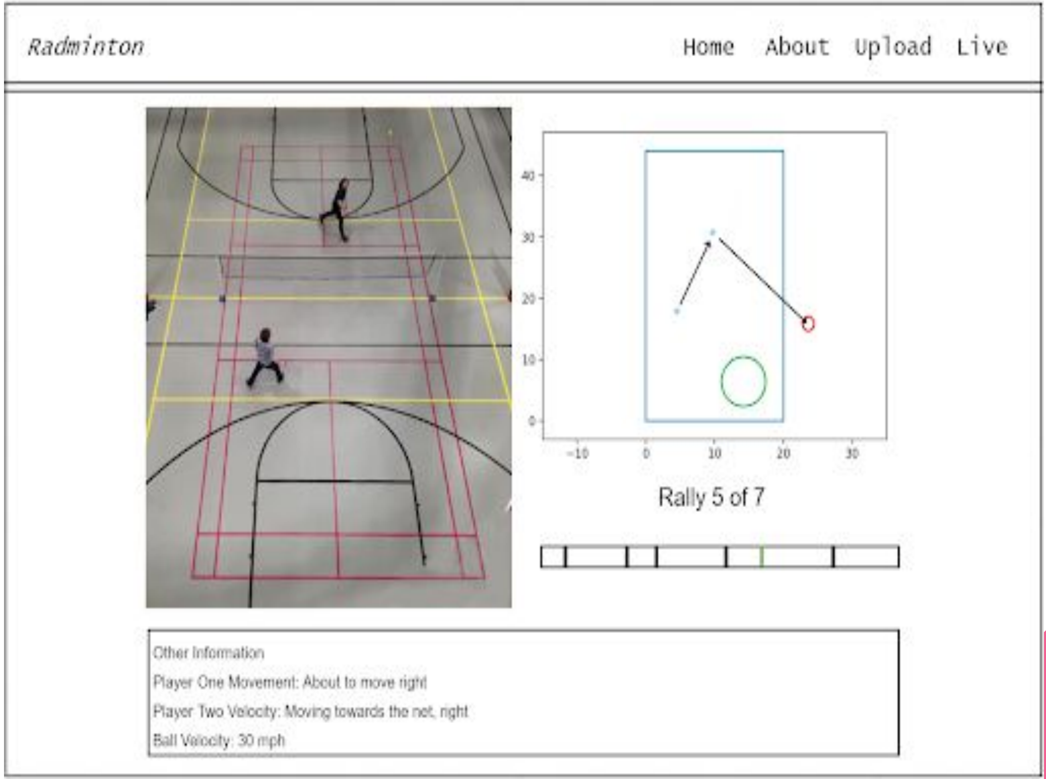
Problem Statement

Currently there are very few software solutions to help people improve in racket sports.

Radminton addresses this problem by providing feedback and suggestions to badminton players of all skill ranges, all through a simple website.

Photo by eric anada from Pexels

Conceptual Sketch



Rally 5 of 7

Other Information

Player One Movement: About to move right

Player Two Velocity: Moving towards the net, right

Ball Velocity: 30 mph

Functional Requirements

- Analyze pre-recorded badminton games and save the data about the game to the server
- Deliver the saved data from the file to the frontend application
- Show that data in a format the user can understand

Constraints and Considerations

01

Position Data Considerations

How to get meaningful shuttlecock data

Limited to 2D coordinates

02

Filming Constraints

How to find a balance between accurate and simple

Some videos do not perform optimally

Use one camera to simplify

03

Maximum Number of Players

Can only support two players

Typical badminton games can be up to four players

Project priorities limited the project to two

Risks and Mitigation

COVID-19

Managed through regular online meetings.

Team was not dependent on physical tools or meetings.

Bad Feedback & Suggestions

Do not want to give players bad feedback.

Mitigated through iterative refinements and filtering.

Poor Data from the User

Low quality recordings limit *Radminton*'s ability to analyze frames.

Limit impact by rejecting videos of low quality and mocking courts.

Resources and Cost Estimation

- Materials
  - Phone to record
  - For our test video: Badminton rackets, shuttlecock
  - Our computers
- Technology
  - IDEs: Pycharm, Webstorm
  - Frameworks/Libraries: OpenCV, ReactJS, etc.

Photo by Vladislav Vasnetsov from Pexels

Cost - $0

- Materials were lent by our client or things we owned
- Technologies were free with ISU licenses or free to public

Project Milestones and Schedule

First Quarter

Identification & Experimentation

Planned the basics

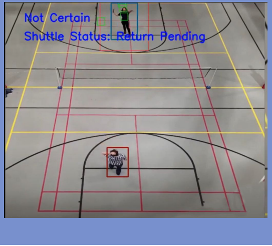
Got basic court, player, and shuttlecock identification done

Second Quarter

Refinement and Demo

Refined identification techniques

Made a demo with debug drawings to visualize our progress



Third Quarter

Web Service Development

Began turning our system into a consumable web service

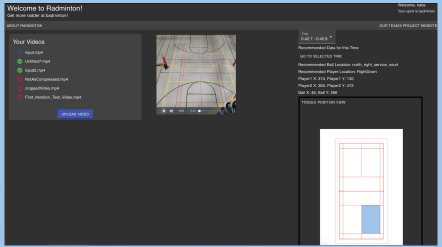
Began developing a web interface

Fourth Quarter

Radminton Alpha

Deployed *Radminton* as a useable web application

Completed work on the web service, website, and suggestions



Welcome to Radminton!
Get more radder at badminton!

Welcome, katie
Your sport is badminton

Your Videos

- input.mp4
- Untitled7.mp4
- input2.mp4
- NotAsCompressed.mp4
- croppedVideo.mp4
- First_Iteration_Test_Video.mp4

Time: 0:42.7 - 0:42.8

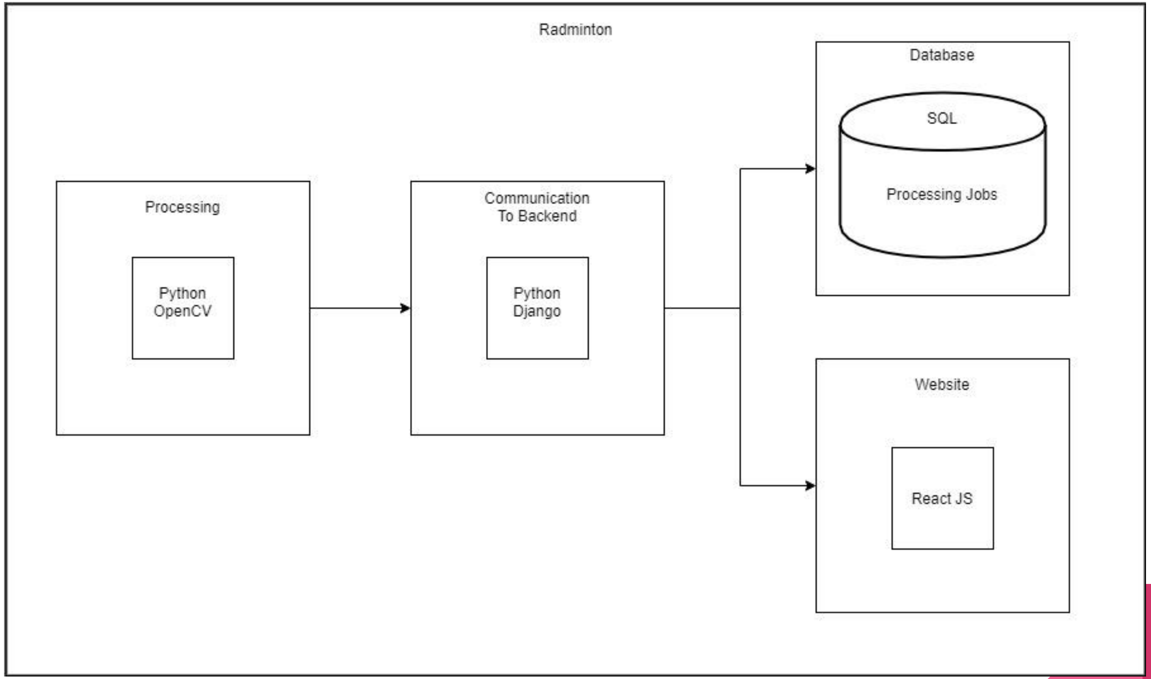
Recommended Data for this Time

Recommended Ball Location: north_right_service_court
Recommended Player Location: RightDown
Player1 X: 310, Player1 Y: 130
Player2 X: 365, Player2 Y: 472
Ball X: 46, Ball Y: 266

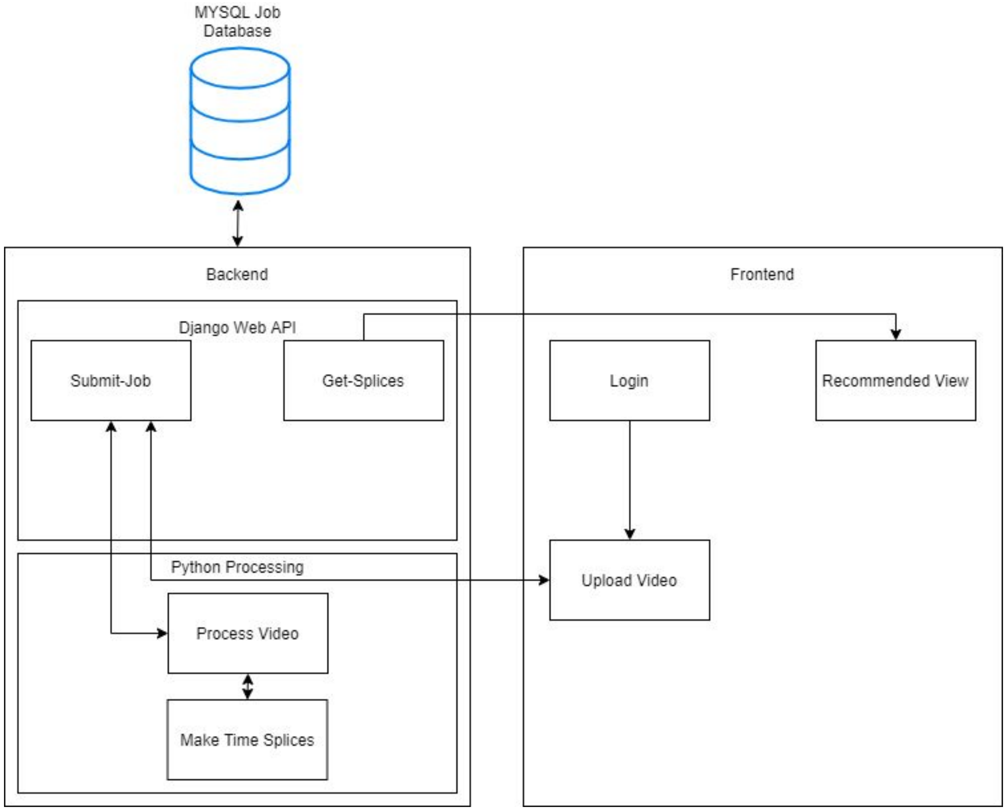
Functional Decomposition

- Identification - Locating the coordinates of each crucial aspect of gameplay.
  - Identify and return court coordinates
  - Identify and track birdie while providing the current coordinates
  - Identify and track the players while returning the coordinates at different parts of their bodies
- Suggestion Engine - Provide quality feedback to user
  - This will inform the user where they could have improved on in different aspects throughout the game.
- Website - Provide easy interface for users to process videos and view processed video data to track their progress over time
  - Login
  - Upload videos to process
  - View past processed videos
  - View recommendation data for videos

Detailed Design

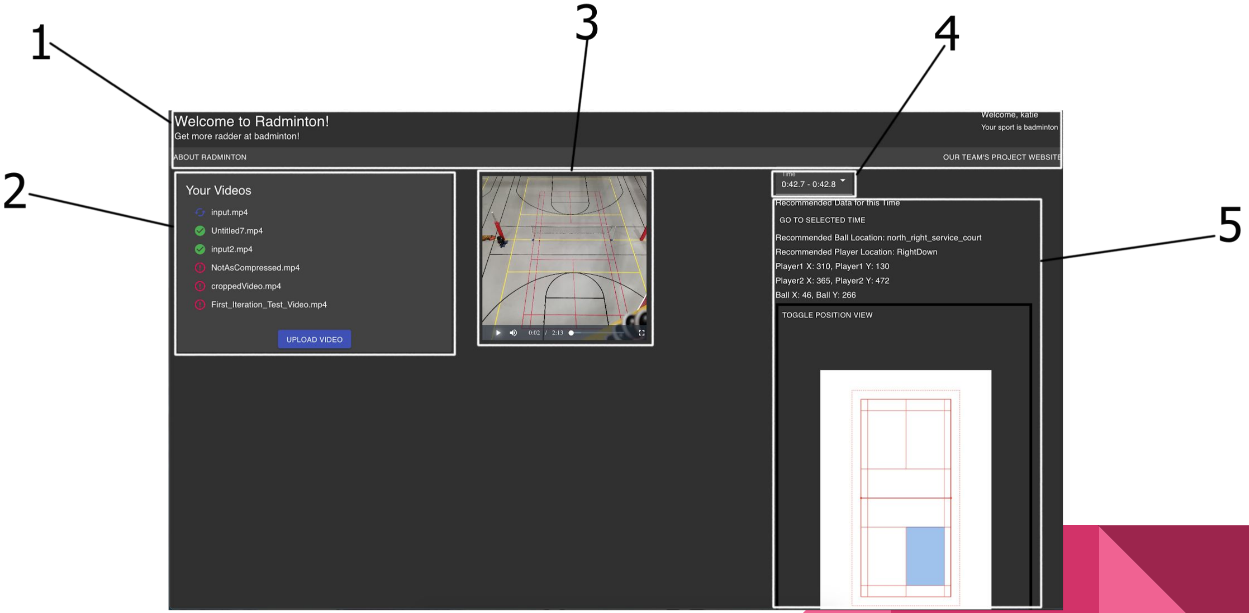


High-Level Architecture



Detailed Design - Frontend

1. Header
2. Videos you have already uploaded, in progress etc., click to select
3. Selected Video
4. Dropdown containing the volley times of selected video
5. Data associated with that volley



Welcome to Radminton!
Get more radder at badminton!

Welcome, katie
Your sport is badminton

Your Videos

- input.mp4
- ✓ Untitled7.mp4
- ✓ input2.mp4
- ! NotAsCompressed.mp4
- ! croppedVideo.mp4
- ! First_Iteration_Test_Video.mp4

0:42.7 - 0:42.8

Recommended Data for this Time

Recommended Ball Location: north_right_service_court
Recommended Player Location: RightDown
Player1 X: 310, Player1 Y: 130
Player2 X: 365, Player2 Y: 472
Ball X: 46, Ball Y: 266

Technology Platforms

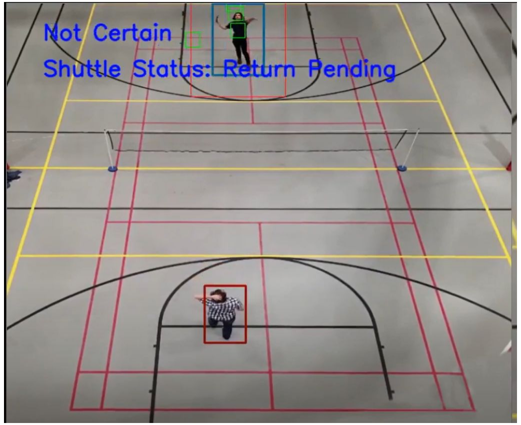
- Python IDE (Pycharm)
- React JS, Redux, Material-UI Components, Axios
- Frontend IDE (Webstorm)
- Django
- OpenCV Framework (Video processing)
- MySQL (Database)

Test Plan

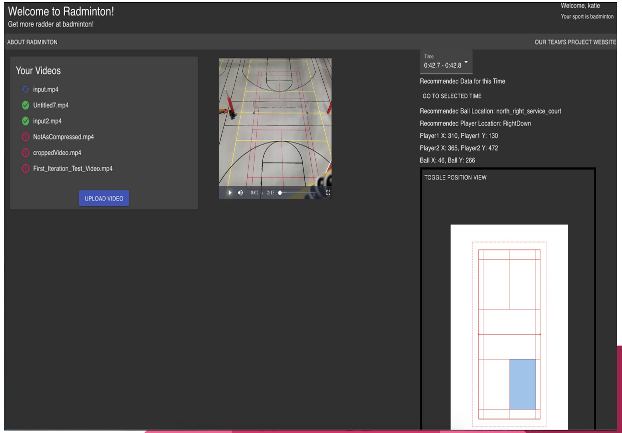
- Python unittest framework
- Testing by breaking the application
- Client evaluation

Prototype Implementations

- Our implementations consisted of processing videos, uploading them, and providing back time splices.



"Insert Empty View"


Welcome to Radminton!

Get more radder at badminton!

Welcome, katie

Your sport is badminton

Your Videos

- input.mp4
- ✓ Untitled7.mp4
- ✓ input2.mp4
- ! NotAsCompressed.mp4
- ! croppedVideo.mp4
- ! First_Iteration_Test_Video.mp4

Time: 0:42.7 - 0:42.8

Recommended Data for this Time

Recommended Ball Location: north_right_service_court

Recommended Player Location: RightDown

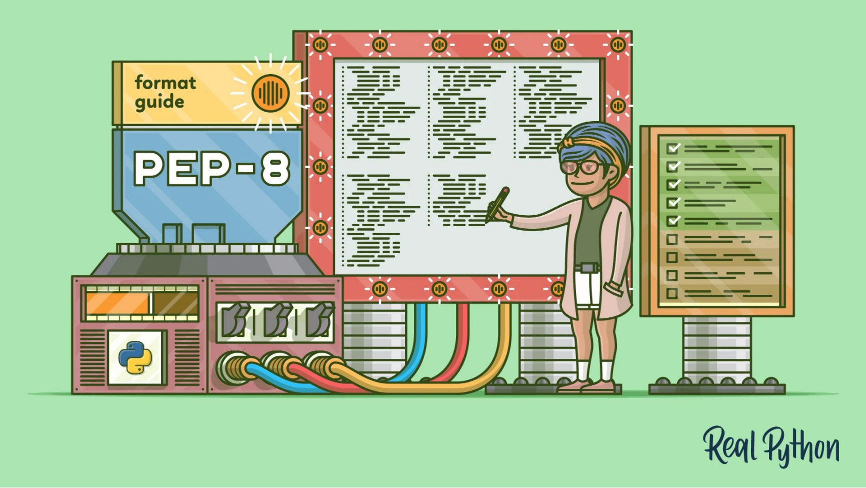
Player1 X: 310, Player1 Y: 130

Player2 X: 365, Player2 Y: 472

Ball X: 46, Ball Y: 266

Engineering Standards and Design Practices

- PEP 8 for Python programming standard to establish consistency across the codebase
- We adhered to standard JavaScript formatting for the React web application



Conclusion

Successes

Top down coordinates of players

Shuttlecock position filtering and analysis

Difficulties

3D shuttlecock coordinates

Shuttlecock tracking

Video quality constraints

What we Learned

OpenCV & frontend technologies

Software architecture & large scale development

Management

Badminton

Contributions

Ben
Shuttlecock tracking, optimization, video player, time skipping, court mocking, CI/CD

Aiden
Header, Recommended View, Test Video Creation

Katie
Real-life coordinates of player, website (initial setup, upload video, video sidebar, processing view, state management of data, login, dropdown with volley time splices)

Brian
Player tracking, court line recognition, web API, database management

John
Hit Detection, Suggestions, Timesplices, initial Django Setup

Christion
Login implementation, CSS techniques to improve aesthetic of website

Future Prospects

Radminton is a part of a much bigger project that aims to provide feedback for a wide variety of racket games. This project was built with the intention that future senior design teams will take the reins and expand on this project in the future.