

# Educational tool for age 10-12 children to enhance language skills and comprehension

---

Group - 24-25J-103

# SUPERVISOR PANNEL



**Ms. Jenny**

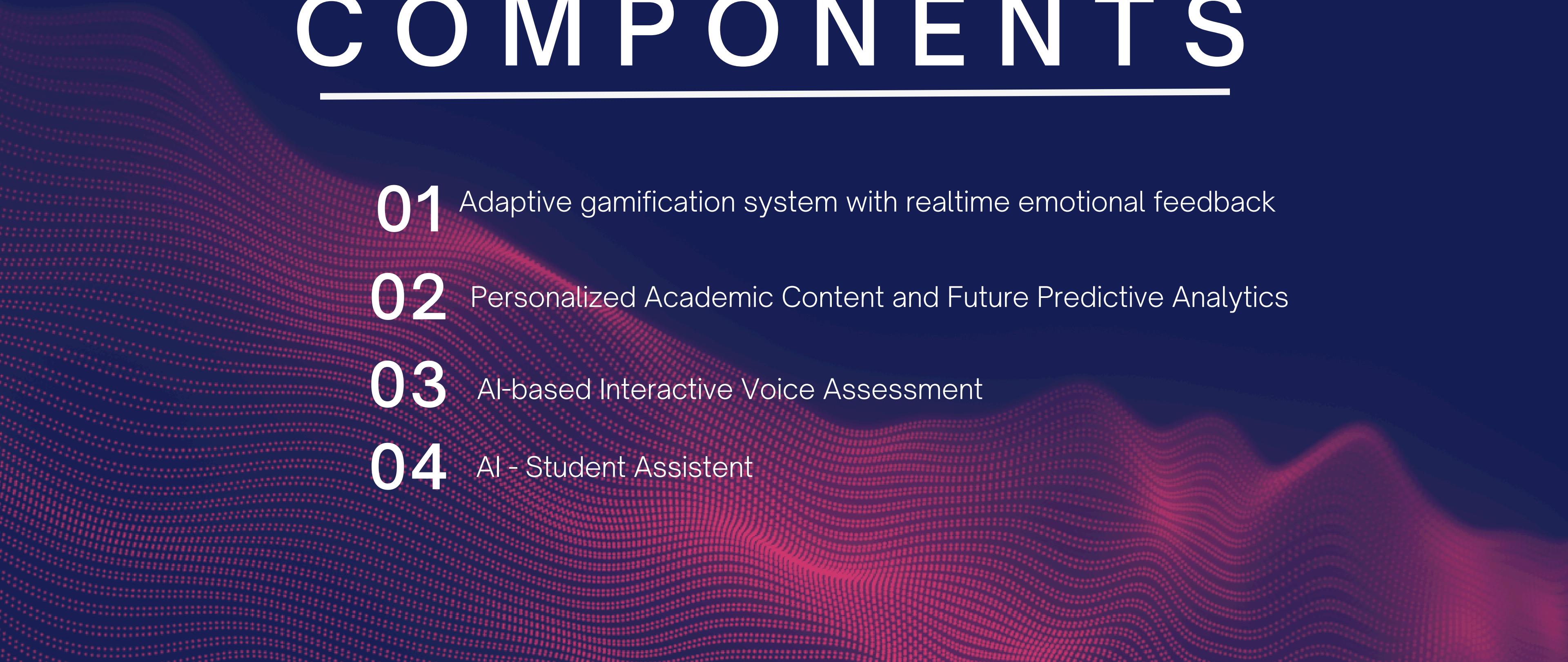
*Supervisor*



**Ms. Dinuka**

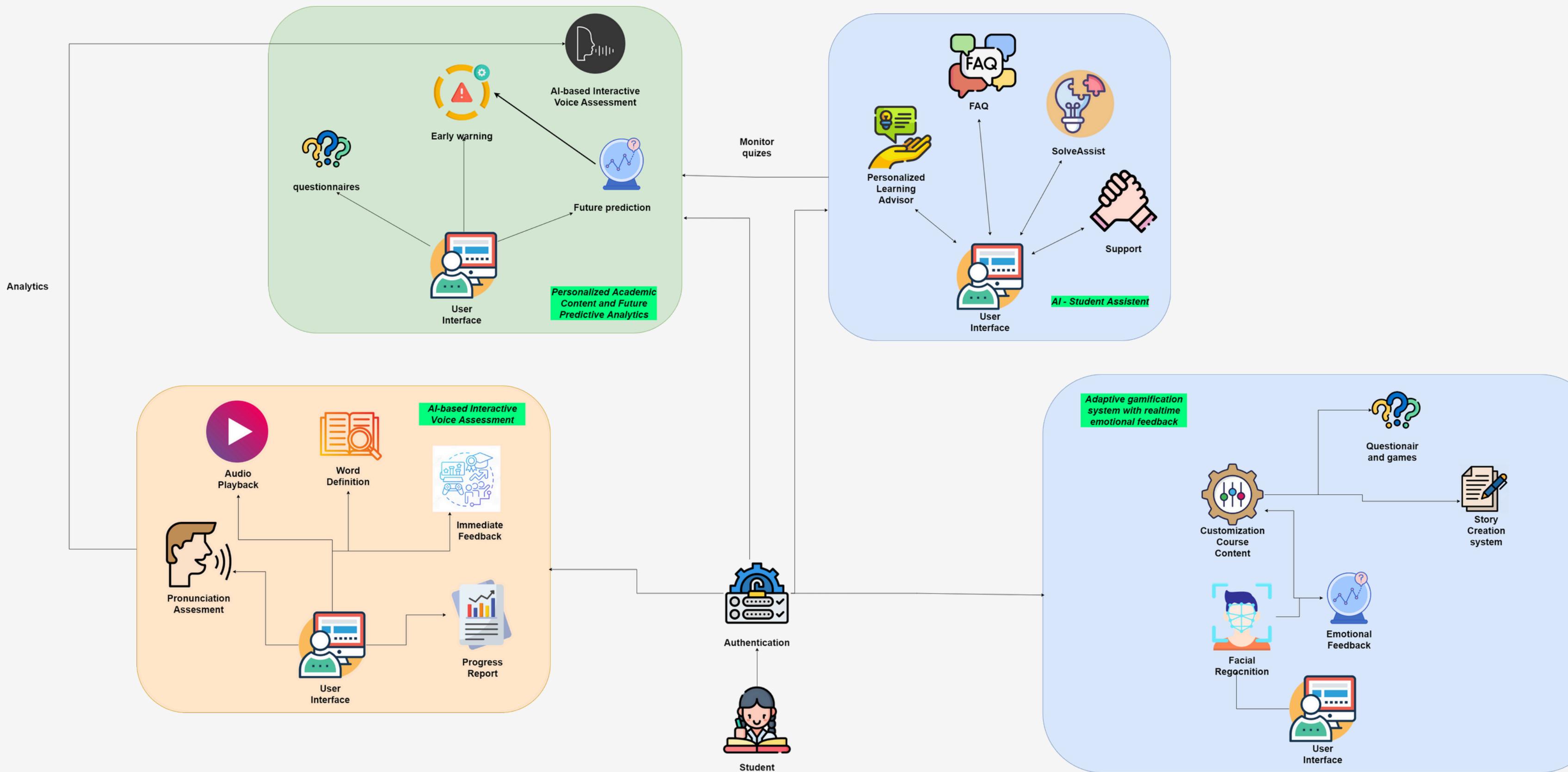
*Co-Supervisor*

# MAIN COMPONENTS



- 01** Adaptive gamification system with realtime emotional feedback
- 02** Personalized Academic Content and Future Predictive Analytics
- 03** AI-based Interactive Voice Assessment
- 04** AI - Student Assistant

# OVERALL SYSTEM DIAGRAMME





# ADAPTIVE GAMIFICATION SYSTEM WITH REALTIME EMOTIONAL FEEDBACK

---



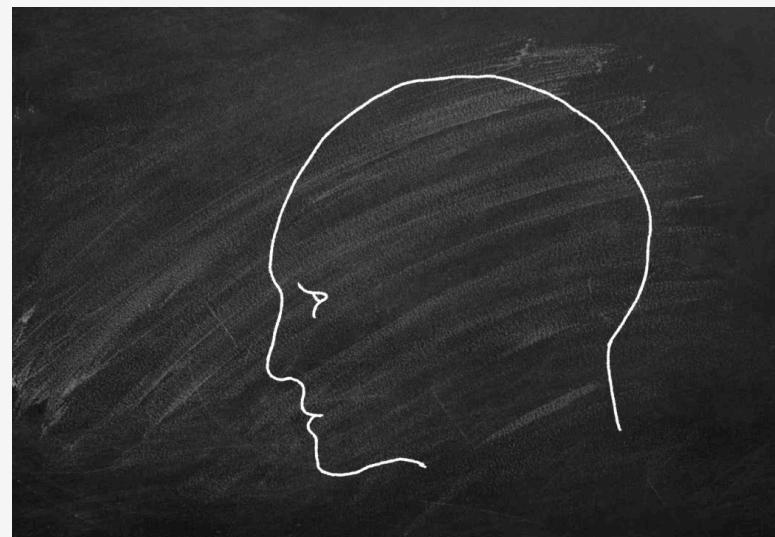
# BACKGROUND

The adaptive gamification system integrates real-time emotional feedback to enhance language skills and comprehension for children aged 10-12. Utilizing AI-based facial recognition, the system detects emotions to dynamically adjust content. Features include interactive story creation, emotion-driven content adaptation, personalized progress reports, and emotional vocabulary expansion, ensuring an engaging and personalized learning experience.

# REASEARCH GAP

Study	Real-time Emotional Feedback	Dynamic Story Creation	Emotion-Driven Content Adaptation	Personalized Progress Reports	Emotional Vocabulary Expansion	Parent and Educator Dashboard
Learning Express	✗	✗	✗	✓	✗	✗
EduVenture	✗	✓	✗	✓	✗	✗
EmotionEd	✓	✗	✗	✗	✗	✓
Proposed System	✓	✓	✓	✓	✓	✓

# RESEARCH PROBLEMS



- How can an adaptive gamification system with real-time emotional feedback enhance language skills and comprehension for children aged 10-12?
- What specific features of gamification can be integrated to effectively engage children in learning activities?
- How can the system be designed to accommodate diverse emotional and learning needs of children within this age group?
- What are the measurable improvements in language skills and comprehension as a result of using this adaptive gamification system?



# SPECIFIC OBJECTIVE

To develop an interactive story creation system that adapts to children's choices and encourages creative thinking and writing practice.



# SUB OBJECTIVES

01

Implement real-time emotional feedback using facial recognition technology to gauge emotions such as joy, confusion, or frustration.

03

Generate personalized progress reports detailing reading and writing progress and emotional engagement.

02

Dynamically adjust story content or introduce mini-games based on the child's emotional state.

04

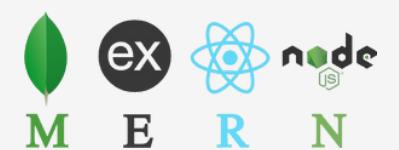
Develop a parent and educator dashboard to provide insights into the child's progress and engagement levels.



# METHODOLOGY

## TECHNOLOGIES

- Natural Language Processing (NLP): SpaCy
- Facial Recognition Technology: Microsoft Azure Face API
- AI-Based Emotional Detection: TensorFlow
- Game Design Principles: Unity
- Data Visualization Tools: Chart.js



# SYSTEM, PERSONAL, AND SOFTWARE REQUIREMENT SPECIFICATION

## Software Requirement

- Visual Studio Code
- Python
- MERN
- Git
- Azure

## Personal Requirement

- School teacher
- School children
- Parents of children
- Child psychologist

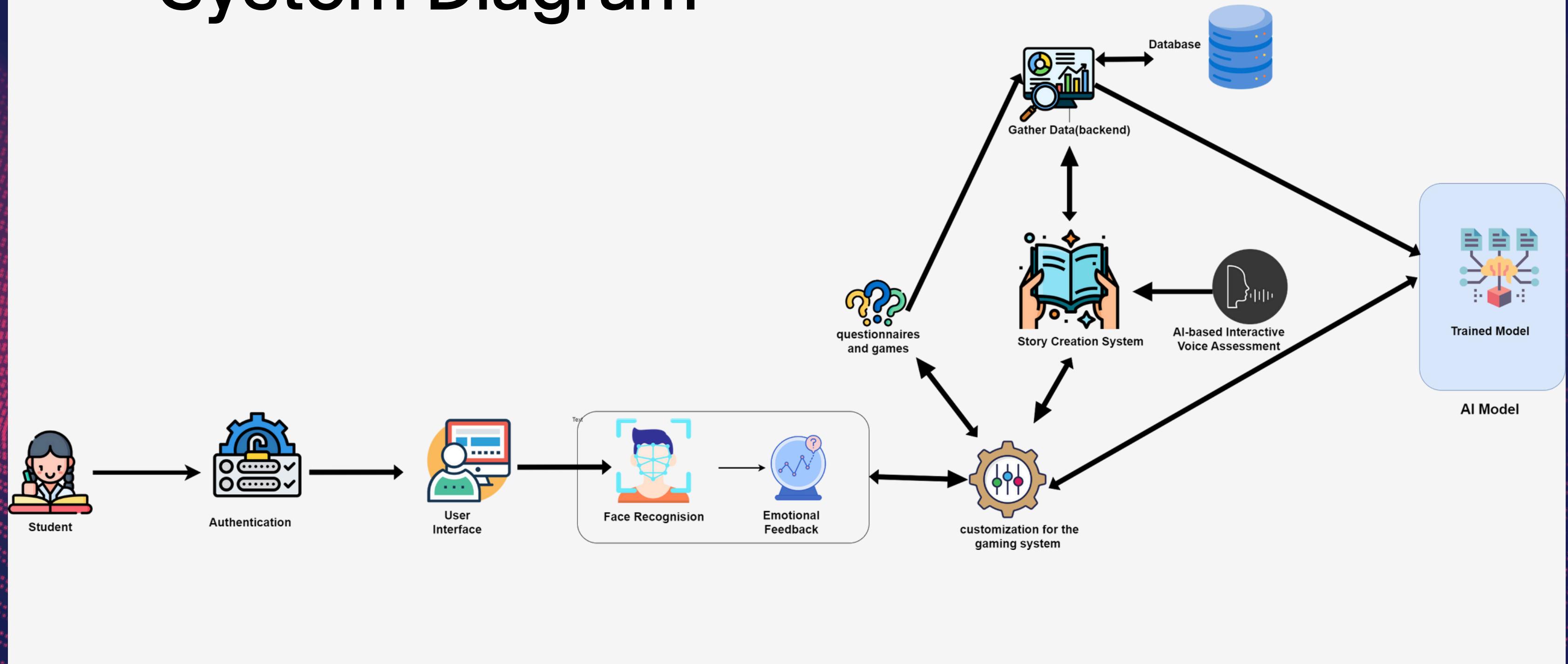
## Functional Requirement

- User Authentication
- Data Collection
- Time Tracking
- Content Personalization
- Real-time Feedback

## Non-Functional Requirement

- Performance
- Reliability and Availability
- The system must be able to handle a large number of requests

# System Diagram





# Data Collection and Annotation

## Data Collection Methods

- From the students at the Regent Language Institute, Negombo
- From the External Supervisor
- online data sets

## Annotation

- From the External Supervisor





The screenshot shows a Google Drive interface. On the left, there's a sidebar with navigation links: Home, My Drive, Computers, Shared with me, Recent, Starred, Spam, Bin, and Storage. It also displays storage usage information: 7.47 GB of 15 GB used, with a 'Get more storage' button. The main area shows a breadcrumb path: Shared with me > ffhq-dataset > images1024x1024. Below this, there are filters for Type, People, and Modified. A list of folders is displayed, each labeled with a three-digit number followed by four zeros: 00000, 01000, 02000, 03000, 04000, 05000, 06000, 07000, 08000, and 09000.

## DATA COLLECTION



## PROOF OF WORK

```
PS C:\Users\Kavishi\Desktop\Research> cd ml
PS C:\Users\Kavishi\Desktop\Research\ml> cd ml
PS C:\Users\Kavishi\Desktop\Research\ml\ml> python -m uvicorn main:app --reload
INFO:     Will watch for changes in these directories: ['C:\\\\Users\\\\Kavishi\\\\Desktop\\\\Research\\\\ml\\\\ml']
INFO:     Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO:     Started reloader process [16016] using StatReload
DEBUG:pymongo.topology:{"topologyId": {"$oid": "67508bd3fab15080ace2db86"}, "message": "Starting topology monitoring"}
DEBUG:pymongo.topology:{"topologyId": {"$oid": "67508bd3fab15080ace2db86"}, "previousDescription": "<TopologyDescription id: 67508bd3fab15080ace2db86, topology_type: Unknown, servers: []>", "newDescription": "<TopologyDescription id: 67508bd3fab15080ace2db86, topology_type: ReplicaSetNoPrimary, servers: [<ServerDescription ('cluster0-shard-00-00.s5foz.mongodb.net', 27017) server_type: Unknown, rtt: None>, <ServerDescription ('cluster0-shard-00-01.s5foz.mongodb.net', 27017) server_type: Unknown, rtt: None>, <ServerDescription ('cluster0-shard-00-02.s5foz.mongodb.net', 27017) server_type: Unknown, rtt: None>]", "message": "Topology description changed"}
DEBUG:pymongo.topology:{"topologyId": {"$oid": "67508bd3fab15080ace2db86"}, "serverHost": "cluster0-shard-00-01.s5foz.mongodb.net", "serverPort": 27017, "message": "Starting server monitoring"}
DEBUG:pymongo.topology:{"topologyId": {"$oid": "67508bd3fab15080ace2db86"}, "serverHost": "cluster0-shard-00-02.s5foz.mongodb.net", "serverPort": 27017, "message": "Starting server monitoring"}
DEBUG:pymongo.topology:{"topologyId": {"$oid": "67508bd3fab15080ace2db86"}, "serverHost": "cluster0-shard-00-00.s5foz.mongodb.net", "serverPort": 27017, "message": "Starting server monitoring"}
INFO:     Started server process [1816]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
```

ML RELOAD



```
PS C:\Users\Kavishi\Desktop\Research> cd web
PS C:\Users\Kavishi\Desktop\Research\web> npm start

Compiled with warnings.

[eslint]
src\App.js
Line 4:8:  'signIn' is defined but never used  no-unused-vars

src\components\nav\Navigation.js
Line 12:18:  'setAvatar' is assigned a value but never used
Line 14:10:  'dropdown' is assigned a value but never used
Line 41:11:  img elements must have an alt prop, either with meaningful text, or an empty string for decorative images  no-unused-vars
                                                     no-unused-vars
                                                     jsx-a11y/alt-text

src\data\navbarItems.js
Line 1:10:  'AiFillInfoCircle' is defined but never used  no-unused-vars
Line 1:28:  'AiOutlineTransaction' is defined but never used  no-unused-vars
Line 1:50:  'AiFillContacts' is defined but never used  no-unused-vars
Line 1:66:  'AiOutlineHeatMap' is defined but never used  no-unused-vars

src\services\Error.Handling.js
Line 1:8:  'React' is defined but never used  no-unused-vars
Line 2:11:  'Redirect' is defined but never used  no-unused-vars

src\services\Users.service.js
Line 97:1:  Assign instance to a variable before exporting as module default  import/no-anonymous-default-export

src\utils\EventEmitter.js
Line 3:1:  Assign instance to a variable before exporting as module default  import/no-anonymous-default-export

Search for the keywords to learn more about each warning.
To ignore, add // eslint-disable-next-line to the line before.

WARNING in [eslint]
src\App.js
Line 4:8:  'SignIn' is defined but never used  no-unused-vars
```

## WEB RELOAD



E-Learning



# Empower Your Learning Regardless the Differences

Discover a smarter way to learn. Our AI-powered platform personalizes your educational experience, helping you reach your goals faster. From curated study plans to interactive content, transform the way you acquire knowledge and skills. Join our community of learners and unlock your full potential today.

Explore Features

Get Started for Free

**WEB RELOAD**



 E-Learning

## Face Identification

Enter Your Username

[Face Identification](#)

Don't have an account? [Sign Up](#)

# LOGIN & REGISTRATION

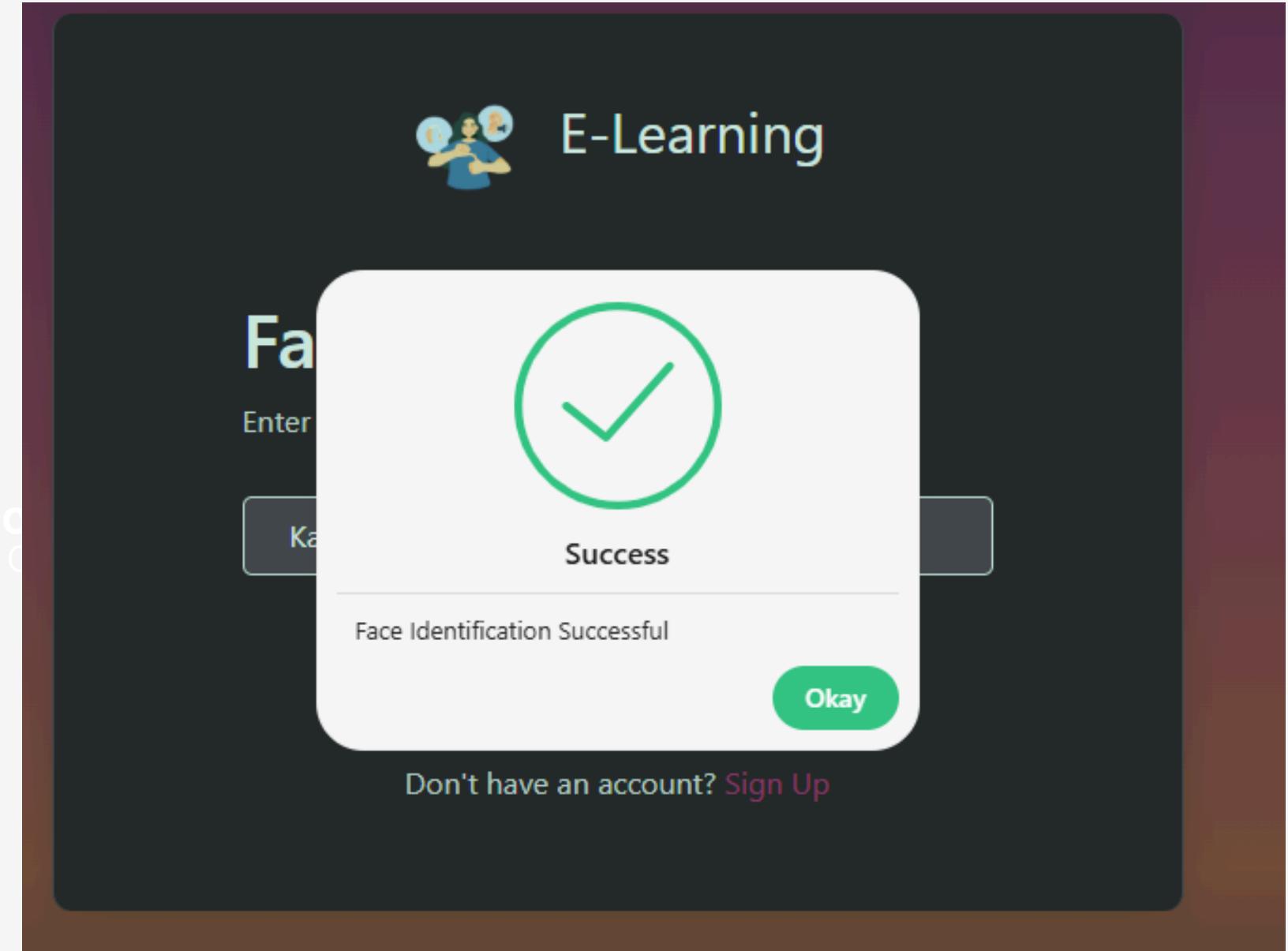
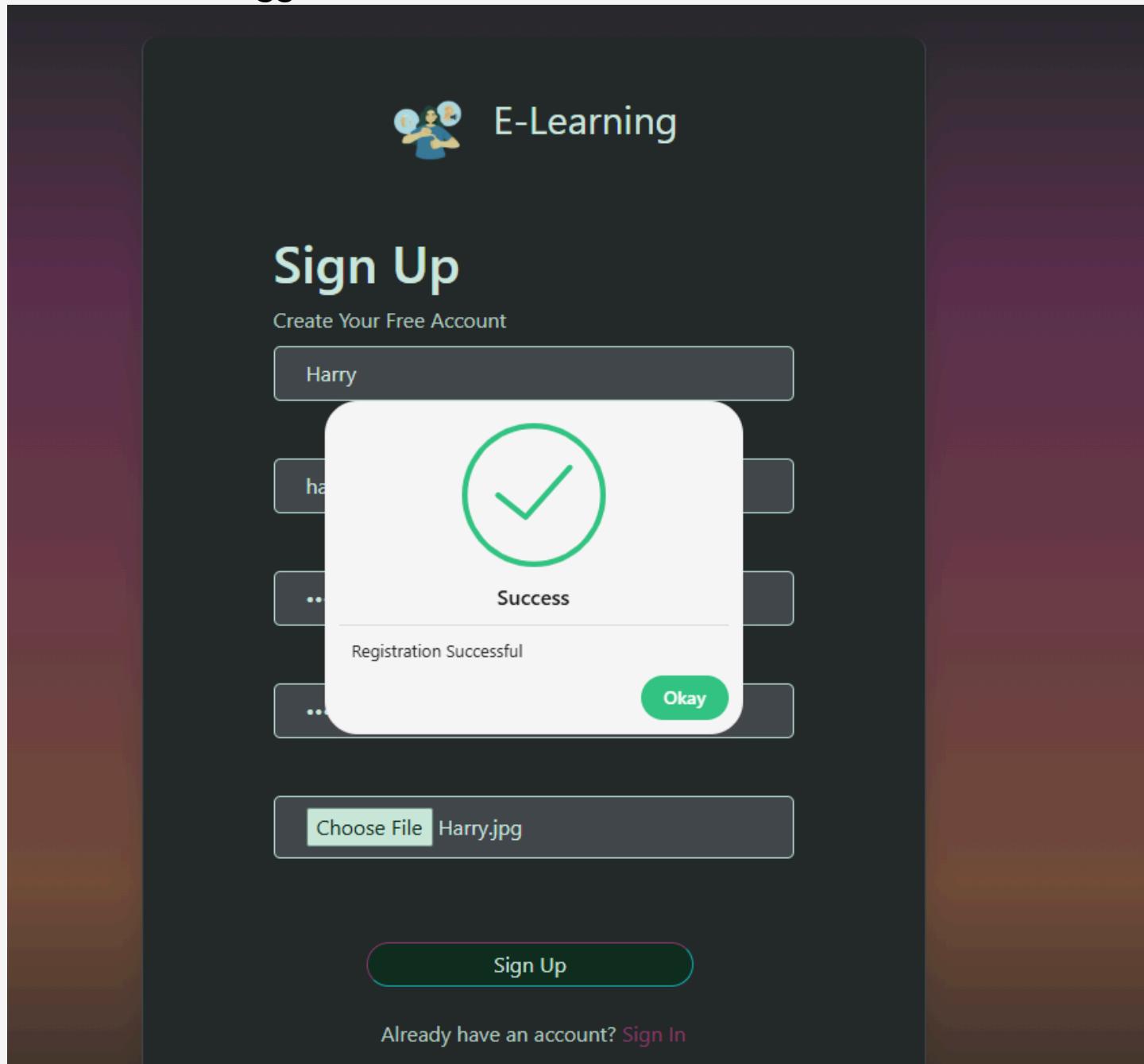
 E-Learning

## Sign Up

Create Your Free Account

[Sign Up](#)

Already have an account? [Sign In](#)



## FACE DETECTION USING FACE NET



```
api_requirements.txt      face_identification.py 1 X
ml > ml > face_identification.py > FaceRecognition > run_recognition
18     class FaceRecognition:
37         def run_recognition(self, input_username):
43             while True:
44                 ret, frame = video_capture.read()
45                 if self.process_current_frame:
46                     small_frame = cv2.resize(frame, (0, 0), fx=0.25, fy=0.25)
47                     rgb_small_frame = cv2.cvtColor(small_frame, cv2.COLOR_BGR2RGB)
48
49
50             # Detect faces
51             self.face_locations = face_recognition.face_locations(rgb_small_frame)
52             self.face_encodings = face_recognition.face_encodings(rgb_small_frame, self.face_locations)
53
54             self.face_names = []
55             for face_encoding in self.face_encodings:
56                 matches = face_recognition.compare_faces(self.known_face_encodings, face_encoding)
57                 name = 'Unknown'
58                 confidence = 'Unknown'
59
60                 face_distances = face_recognition.face_distance(self.known_face_encodings, face_encoding)
61                 best_match_index = np.argmin(face_distances)
62
63                 if matches[best_match_index]:
64                     name = self.known_face_names[best_match_index]
65                     confidence = face_confidence(face_distances[best_match_index])
66
67                     # Extract the base name (without extension) for comparison
68                     recognized_name = os.path.splitext(name)[0]
69                     self.face_names.append(f'{recognized_name} ({confidence})')
70                     print(recognized_name+'---'+input_username)
71
72                     # Compare with the input username
73                     if recognized_name == input_username:
74                         detected = True
75                         break # Exit the Loop if a match is found
```

## ACCURACY CHECKING



## COMPLETED COMPONENTS 50%

Data Collection  
ML Algorithm development for face net  
Image recognition using tensorflow  
video capturing using open cv  
Data saving for face registration  
Face Accuracy checking for users

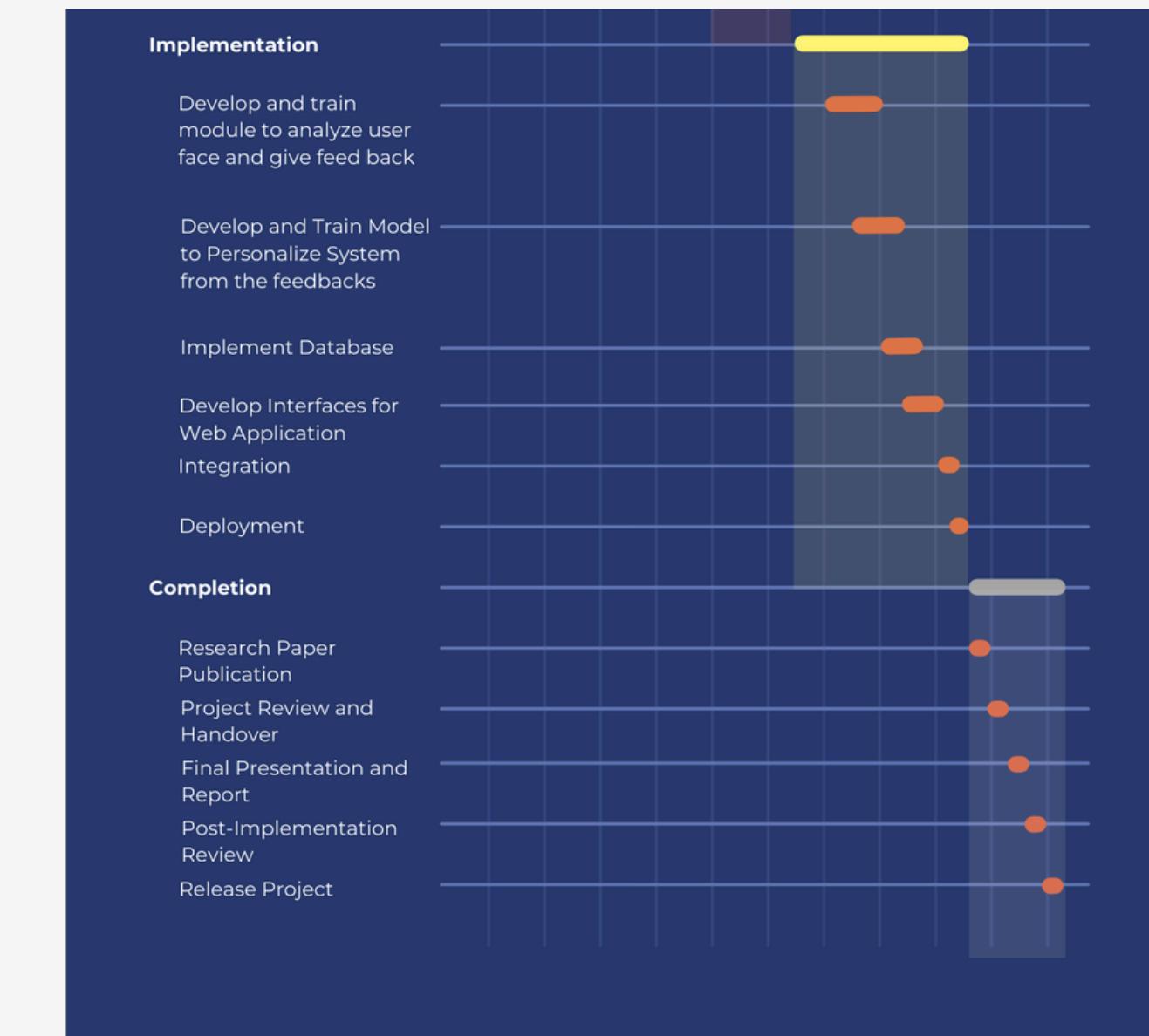
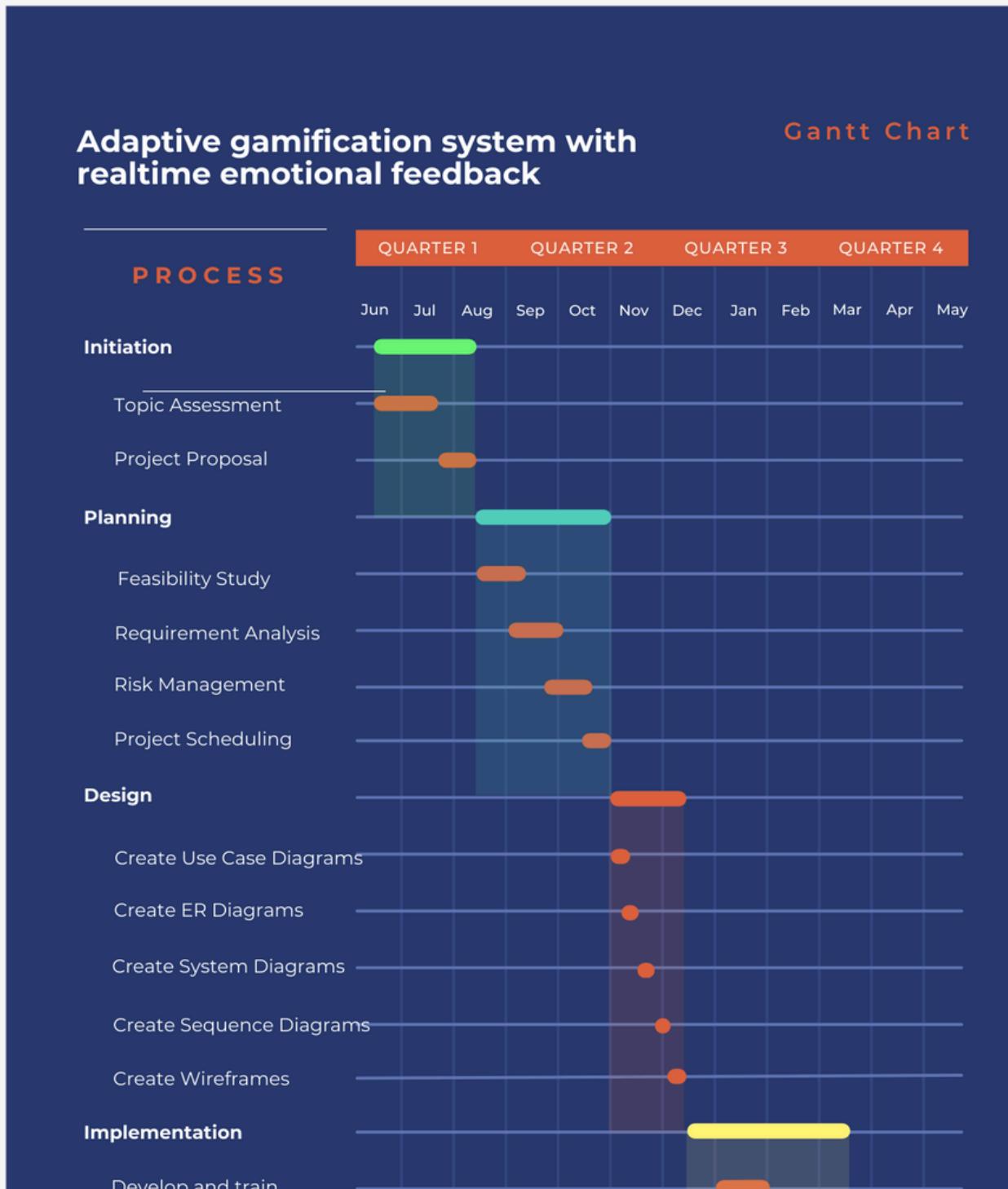
DATA COLLECTION  
FACE NET IMAGE RECOGNITION USING TENSORFLOW  
VIDEO CAPTURING USING OPEN CV  
DATA SAVING FOR FACE REGISTRATION  
FACE ACCURACY CHECKING FOR USERS

## FUTURE DEVELOPMENT 50%

Emotion Detection  
Games system development  
feedback management system with  
personalized dashboard



# GANNT CHART



# REFERENCES

- D'Mello, S. K., & Kory, J. (2015). A review and meta-analysis of multimodal affect detection systems. *ACM Computing Surveys (CSUR)*, 47(3), 1-36.
- Baker, R. S., D'Mello, S. K., Rodrigo, M. M. T., & Graesser, A. C. (2012). Better to be frustrated than bored: The incidence, persistence, and impact of learners' cognitive-affective states during interactions with three different computer-based learning environments. *International Journal of Human-Computer Studies*, 68(4), 223-241.
- Hattie, J., & Timperley, H. (2007). The power of feedback. *Review of Educational Research*, 77(1), 81- 112.
- Brackett, M. A., & Rivers, S. E. (2014). Transforming students' lives with social and emotional learning. *International Handbook of Emotions in Education*, 368-388.
- Desforges, C., & Abouchaar, A. (2003). The impact of parental involvement, parental support, and family education on pupil achievements and adjustment: A literature review. Department for Education and Skills.
- Hirsh-Pasek, K., Zosh, J. M., Golinkoff, R. M., Gray, J. H., Robb, M. B., & Kaufman, J. (2015). Putting education in “educational” apps: Lessons from the science of learning. *Psychological Science in the Public Interest*, 16(1), 3-34.



# PERSONALIZED ACADEMIC CONTENT AND FUTURE PREDICTIVE ANALYTICS

---



# BACKGROUND

- Existing educational systems leverage machine learning and personalized content but often lack real-time adaptability and early intervention predictive analytics.
- Most systems adjust educational content based on current performance metrics without robust forecasting capabilities.
- Additionally, existing systems broadly focus on multiple age groups and subjects, often overlooking specific needs of Academic English learners aged 10-12.

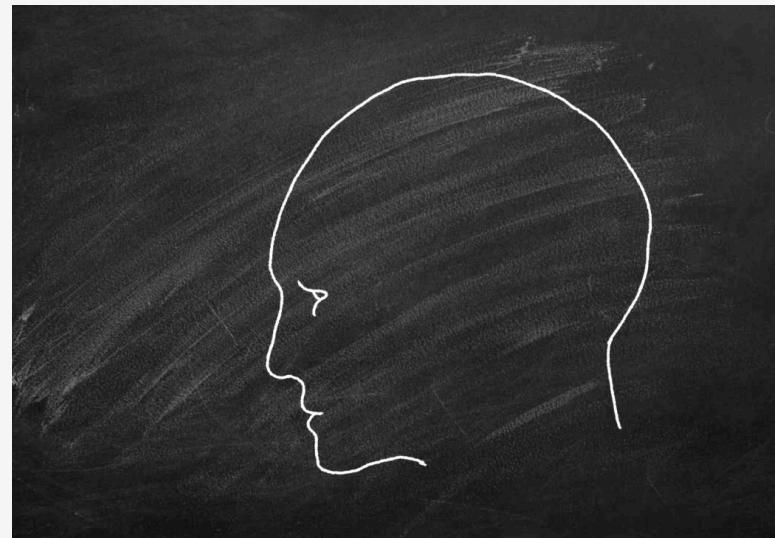
# REASEARCH GAP

Project 1 - Akçapınar, G., Hasnine, M. N., Majumdar, R., Flanagan, B., Ogata, H. (2019). Developing an earlywarning system for spotting at-risk students by using eBook interaction logs.

Project 2 - Nimy, E., Mosia, M., Chibaya, C. (2023). Identifying At-Risk Students for Early Intervention  
 project 3 - Knewton (existing System)

Study	Customize Course content	NLP answer analyzer	Future Predictive Analytics	Early Warning Risk Student	Alignment with School Syllabus
Project 1	✗	✗	✗	✓	✗
Project 2	✗	✗	✓	✓	✗
Project 3	✓	✗	✗	✗	✗
Proposed System	✓	✓	✓	✓	✓

# RESEARCH PROBLEMS



- How does personalized content impact learning outcomes in children aged 10-12?  
addressing individual strengths and weaknesses, promoting better engagement
- Can predictive analytics effectively identify at-risk students before noticeable declines in performance  
analyzing patterns and predicting trends, spot who might need extra help
- What role does real-time data play in educational content adaptation?  
immediate adjustments to the learning material, each student needs at that moment
- What advantages does dynamic difficulty scaling offer over traditional static content delivery?  
provides challenges appropriate to the student's current capability



# SUB OBJECTIVES

**01**

Develop and implement a data collection framework that respects privacy and accurately captures necessary metrics and NLP model

**02**

Create predictive models that forecast learning trajectories and identify needs for early intervention.

**03**

Design personalized learning content that adapts dynamically to the student's progress.

**04**

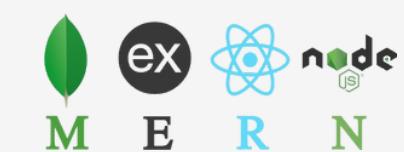
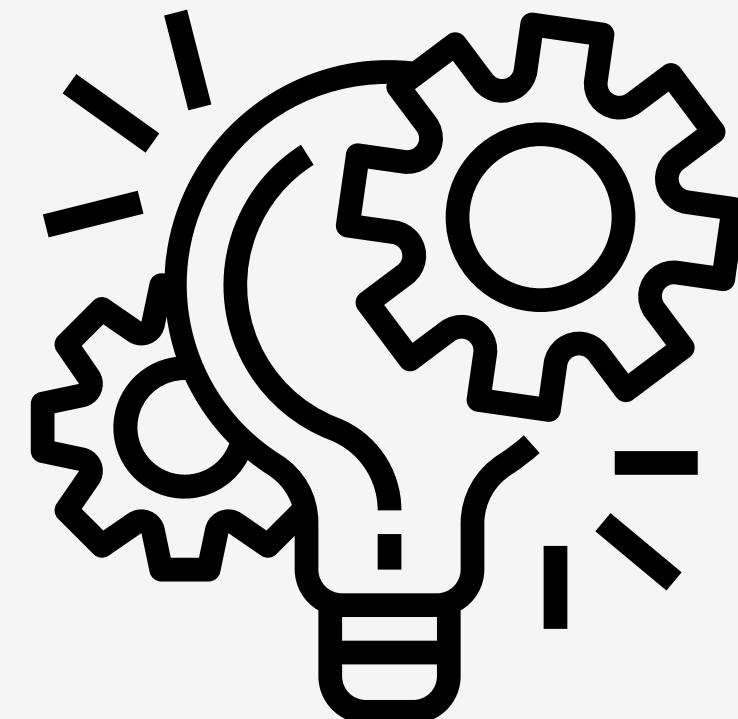
Implement real-time content adaptability, ensuring that the educational material is continuously aligned with the student's current needs.



# METHODOLOGY

## TECHNOLOGIES

MERN  
Tensorflow  
Python  
AWS/Google Cloud/Azure  
Git



# SYSTEM, PERSONAL, AND SOFTWARE REQUIREMENT SPECIFICATION

## Software Requirement

- Tensorflow
- Python
- MERN
- VsCode

## Personal Requirement

- School teacher
- School children
- Parents of children

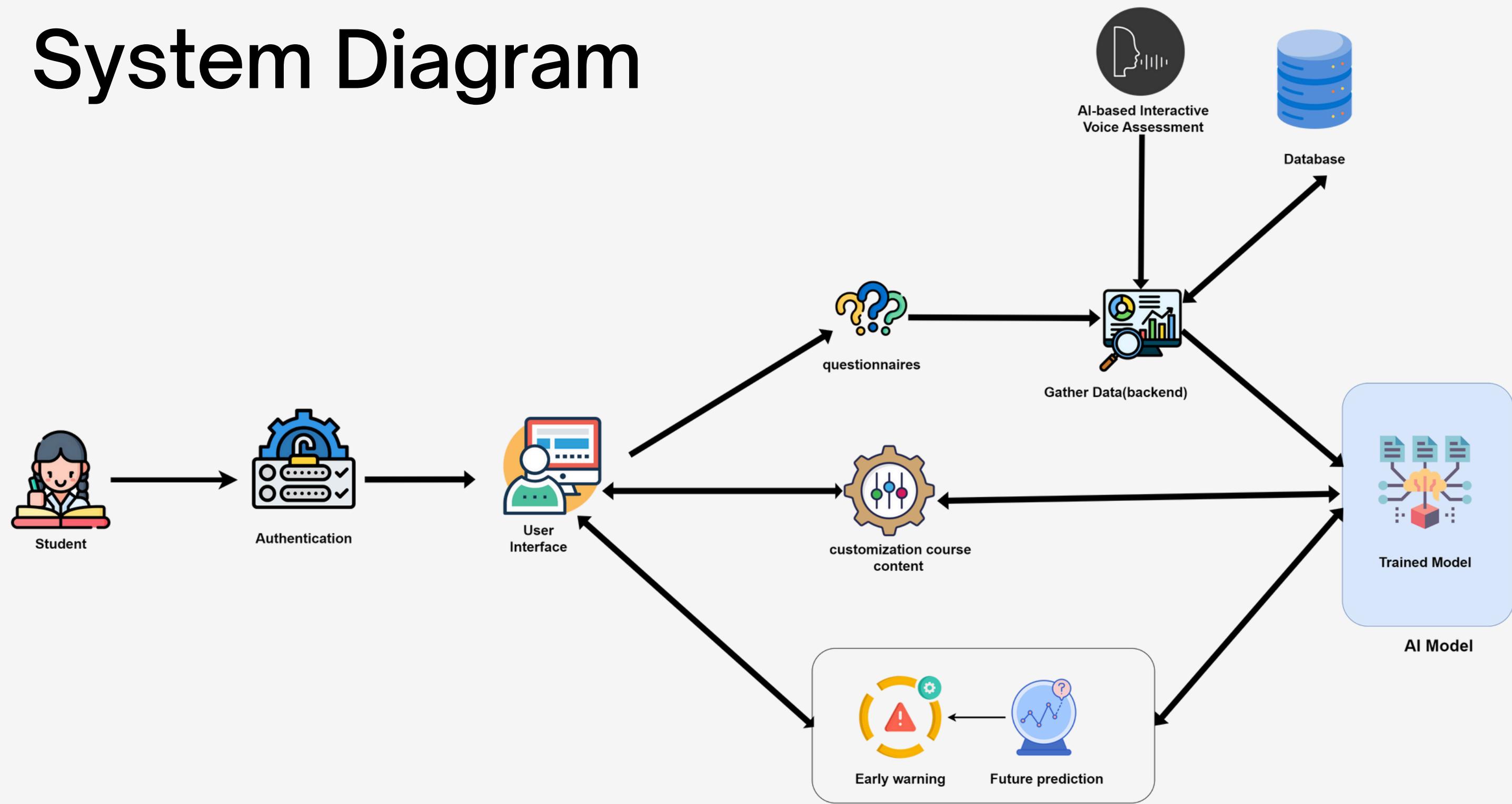
## Functional Requirement

- User Authentication
- Data Collection
- Predictive Analytics
- Content Personalization
- Real-time Feedback

## Non-Functional Requirement

- Performance
- Reliability and Availability
- The system must be able to handle a large number of requests

# System Diagram





# Data Collection and Annotation

## Data Collection Methods

- From the students at the Regent Language Institute, Negombo
- From the External Supervisor

## Annotation

- From the External Supervisor





# NLP Model For Question Analyze

# Proof of work

# When starting

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

D:\Sachintha\ResearchProject\venv\lib\site-packages\transformers\optimization.py:411: FutureWarning: This implementation of AdamW is deprecated and will be removed in v4.0.0. Please use the new implementation in the transformers package.  
his warning  
{'eval\_loss': 0.587438702583313, 'eval\_runtime': 0.3989, 'eval\_samples\_per\_second': 5.013, 'eval\_steps\_per\_second': 2.507, 'epoch': 1.0}  
{'eval\_loss': 0.5850461721420288, 'eval\_runtime': 0.377, 'eval\_samples\_per\_second': 5.305, 'eval\_steps\_per\_second': 2.653, 'epoch': 3.0}  
{'train\_runtime': 50.8479, 'train\_samples\_per\_second': 0.649, 'train\_steps\_per\_second': 0.118, 'train\_loss': 0.694302479426066, 'epoch': 3.0}  
100% | 6/6 [00:50<00:00, 8.47s/it]  
Model and tokenizer saved to './models/'  
Evaluation Accuracy: 0.00%  
Evaluation Accuracy: 0.00%  
(venv) PS D:\Sachintha\ResearchProject> python scripts/prediction.py  
Logits: tensor([[-0.9039, -0.2275]])  
Probabilities: tensor([[0.3371, 0.6629]])  
Similarity Score: 0.6629  
Prediction: Incorrect, Similarity Score: 0.6629  
(venv) PS D:\Sachintha\ResearchProject> python scripts/evaluation.py  
Evaluation Accuracy: 0.00%



# Proof of work and Progress

```
You should probably train this model on a down-stream task to be able to use it for predictions and inference.
D:\Sachintha\ResearchProject\venv\lib\site-packages\transformers\optimization.py:411: FutureWarning: This implementation of AdamW is deprecated and will be removed in a future version. Use the PyTorch implementation torch.optim.AdamW instead, or set `no_deprecation_warning=True` to disable this warning
{'eval_loss': 0.6667951345443726, 'eval_runtime': 6.6861, 'eval_samples_per_second': 5.534, 'eval_steps_per_second': 0.748, 'epoch': 1.0}
{'eval_loss': 0.6711542010307312, 'eval_runtime': 6.8857, 'eval_samples_per_second': 5.373, 'eval_steps_per_second': 0.726, 'epoch': 2.0}
{'eval_loss': 0.6708850264549255, 'eval_runtime': 6.3859, 'eval_samples_per_second': 5.794, 'eval_steps_per_second': 0.783, 'epoch': 3.0}
100%|██████████| 66/66 [05:10<00:00, 4.71s/it]
Model and tokenizer saved to './models/'

(venv) PS D:\Sachintha\ResearchProject> python scripts/evaluation.py
Evaluation Accuracy: 36.84%
Logits: tensor([[ 0.5492, -0.8669]])
Probabilities: tensor([[0.8044, 0.1956]])
Similarity Score: 0.1956
Prediction: Incorrect, Similarity Score: 0.1956
```

Phase 1

```
d will be removed in a future version. Use the PyTorch implementation torch.optim.AdamW instead, or set `no_deprecation_warning=True` to disable this warning
warnings.warn(
{'eval_loss': 0.6744022369384766, 'eval_runtime': 6.1825, 'eval_samples_per_second': 5.985, 'eval_steps_per_second': 0.809, 'epoch': 1.0}
{'eval_loss': 0.6687902212142944, 'eval_runtime': 6.4128, 'eval_samples_per_second': 5.77, 'eval_steps_per_second': 0.78, 'epoch': 2.0}
{'eval_loss': 0.7073997855186462, 'eval_runtime': 6.2682, 'eval_samples_per_second': 5.903, 'eval_steps_per_second': 0.798, 'epoch': 4.0}
{'eval_loss': 0.719953179359436, 'eval_runtime': 6.4198, 'eval_samples_per_second': 5.763, 'eval_steps_per_second': 0.779, 'epoch': 5.0}
{'eval_loss': 0.7281244993209839, 'eval_runtime': 6.365, 'eval_samples_per_second': 5.813, 'eval_steps_per_second': 0.786, 'epoch': 6.0}
100%|██████████| 66/66 [09:23<00:00, 8.53s/it]
Model and tokenizer saved to './models/'

(venv) PS D:\Sachintha\ResearchProject> python scripts/evaluation.py
Evaluation Accuracy: 52.63%
Logits: tensor([[-0.3802, -0.022911]])
Probabilities: tensor([[0.4116, 0.5884]])
Similarity Score: 0.5884
```



# Proof of work and Progress

```
D:\Sachintha\ResearchProject\venv\lib\site-packages\transformers\optimization.py:411: FutureWarning: This implementation of AdamW is deprecated and will be removed in v4.0.0. Please use the PyTorch implementation of the AdamW optimizer instead.
  warnings.warn(
{'eval_loss': 0.6804381012916565, 'eval_runtime': 12.7968, 'eval_samples_per_second': 5.861, 'eval_steps_per_second': 0.781, 'epoch': 1.0}
{'eval_loss': 0.6648784875869751, 'eval_runtime': 12.7818, 'eval_samples_per_second': 5.868, 'eval_steps_per_second': 0.782, 'epoch': 2.0}
{'eval_loss': 0.5316901206970215, 'eval_runtime': 12.6222, 'eval_samples_per_second': 5.942, 'eval_steps_per_second': 0.792, 'epoch': 4.0}
{'eval_loss': 0.46550628542900085, 'eval_runtime': 12.5471, 'eval_samples_per_second': 5.977, 'eval_steps_per_second': 0.797, 'epoch': 5.0}
{'eval_loss': 0.4325026571750641, 'eval_runtime': 12.5213, 'eval_samples_per_second': 5.99, 'eval_steps_per_second': 0.799, 'epoch': 6.0}
{'train_runtime': 1081.9449, 'train_samples_per_second': 1.941, 'train_steps_per_second': 0.122, 'train_loss': 0.5660151857318301, 'epoch': 6.0}
Model and tokenizer saved to './models/'

(venv) PS D:\Sachintha\ResearchProject> python scripts/evaluation.py
Evaluation Accuracy: 82.67%
(venv) PS D:\Sachintha\ResearchProject> python scripts/prediction.py
Probabilities: tensor([[0.2822, 0.7178]])
Similarity Score: 0.7178
```

```
warnings.warn(
{'eval_loss': 0.5925895571708679, 'eval_runtime': 7.2846, 'eval_samples_per_second': 12.767, 'eval_steps_per_second': 1.647, 'epoch': 1.0}
{'eval_loss': 0.39768004417419434, 'eval_runtime': 7.2936, 'eval_samples_per_second': 12.751, 'eval_steps_per_second': 1.645, 'epoch': 2.0}
{'eval_loss': 0.32142123579978943, 'eval_runtime': 7.1069, 'eval_samples_per_second': 13.086, 'eval_steps_per_second': 1.688, 'epoch': 3.0}
{'eval_loss': 0.3134922981262207, 'eval_runtime': 6.823, 'eval_samples_per_second': 13.63, 'eval_steps_per_second': 1.759, 'epoch': 4.0}
{'eval_loss': 0.1405254602432251, 'eval_runtime': 6.9687, 'eval_samples_per_second': 13.345, 'eval_steps_per_second': 1.722, 'epoch': 5.0}
{'eval_loss': 0.2282506674528122, 'eval_runtime': 7.3123, 'eval_samples_per_second': 12.718, 'eval_steps_per_second': 1.641, 'epoch': 6.0}
{'train_runtime': 769.3663, 'train_samples_per_second': 3.4, 'train_steps_per_second': 0.218, 'train_loss': 0.3163152876354399, 'epoch': 6.0}
100%]
Model and tokenizer saved to './models/'

(venv) PS C:\Git-Hub-Repos\ResearchProject> python scripts/evaluation.py
Evaluation Accuracy: 95.74%
```



# Prediction for answer Correction

```
(venv) PS C:\Git-Hub-Repo\ResearchProject> python scripts/prediction.py
Logits: tensor([[-2.8079,  2.6700]])
Probabilities: tensor([[0.0942, 0.9958]])
Similarity Score: 0.9958
Prediction: Correct, Similarity Score: 0.9958
(venv) PS C:\Git-Hub-Repo\ResearchProject> python scripts/prediction.py
Logits: tensor([[ 2.4783, -2.3661]])
Probabilities: tensor([[0.9922, 0.0078]])
Similarity Score: 0.0078
Prediction: Incorrect, Similarity Score: 0.0078
(venv) PS C:\Git-Hub-Repo\ResearchProject> python scripts/prediction.py
Logits: tensor([[-2.8990,  2.6530]])
Probabilities: tensor([[0.0039, 0.9961]])
Similarity Score: 0.9961
Prediction: Correct, Similarity Score: 0.9961
(venv) PS C:\Git-Hub-Repo\ResearchProject> python scripts/prediction.py
Logits: tensor([[ 2.5107, -2.0807]])
Probabilities: tensor([[0.9901, 0.0099]])
Similarity Score: 0.0099
Prediction: Incorrect, Similarity Score: 0.0099
(venv) PS C:\Git-Hub-Repo\ResearchProject> python scripts/prediction.py
Logits: tensor([[-1.8023,  2.3155]])
Probabilities: tensor([[0.0166, 0.9840]])
Similarity Score: 0.9840
Prediction: Correct, Similarity Score: 0.9840
(venv) PS C:\Git-Hub-Repo\ResearchProject> []
```

for Trained data

for Untrained data



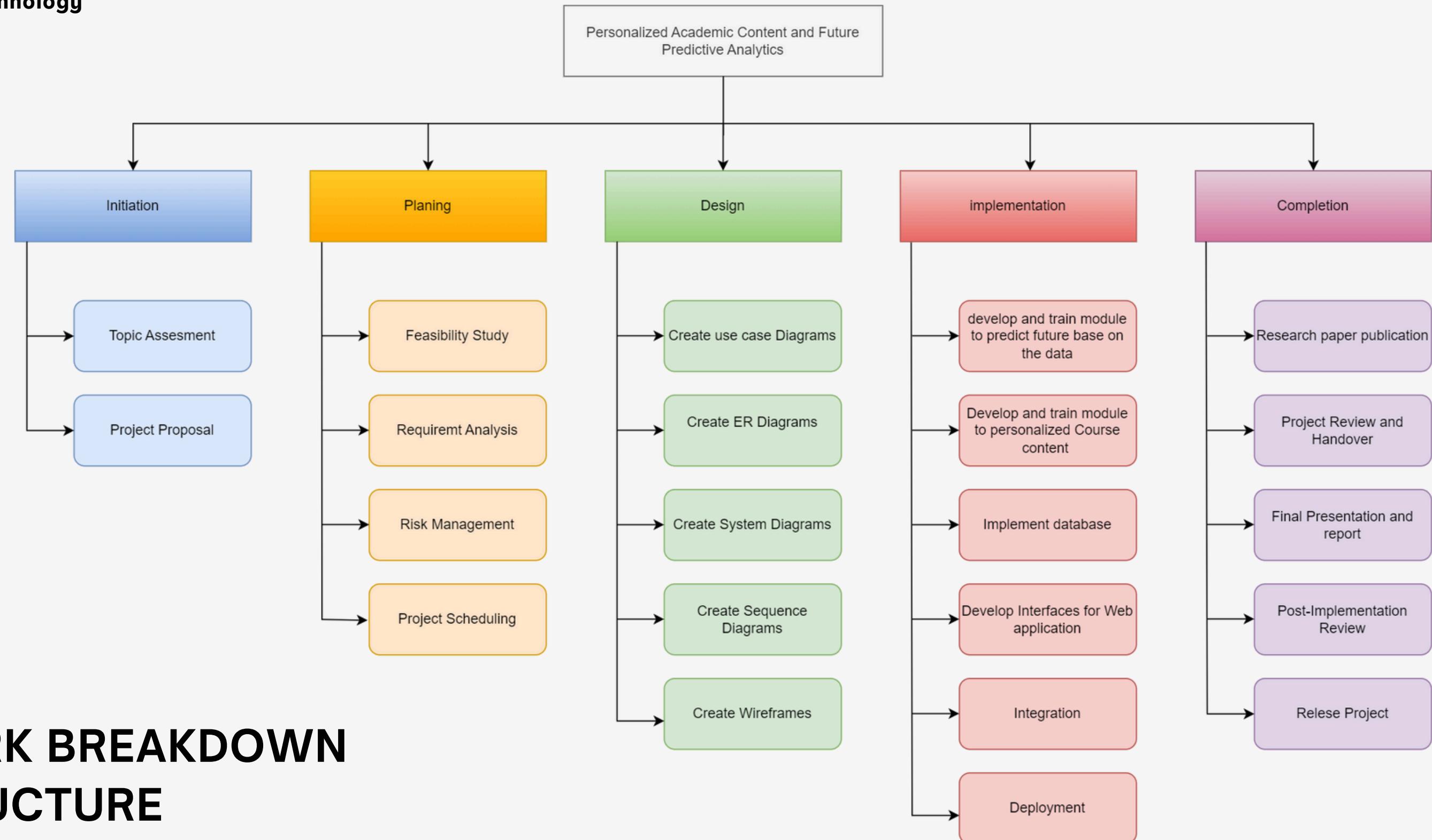
66

## COMPLETED COMPONENTS

- Data Collection - 50%
- NLP Model for get answer correction for paper

## FUTURE DEVELOPMENT

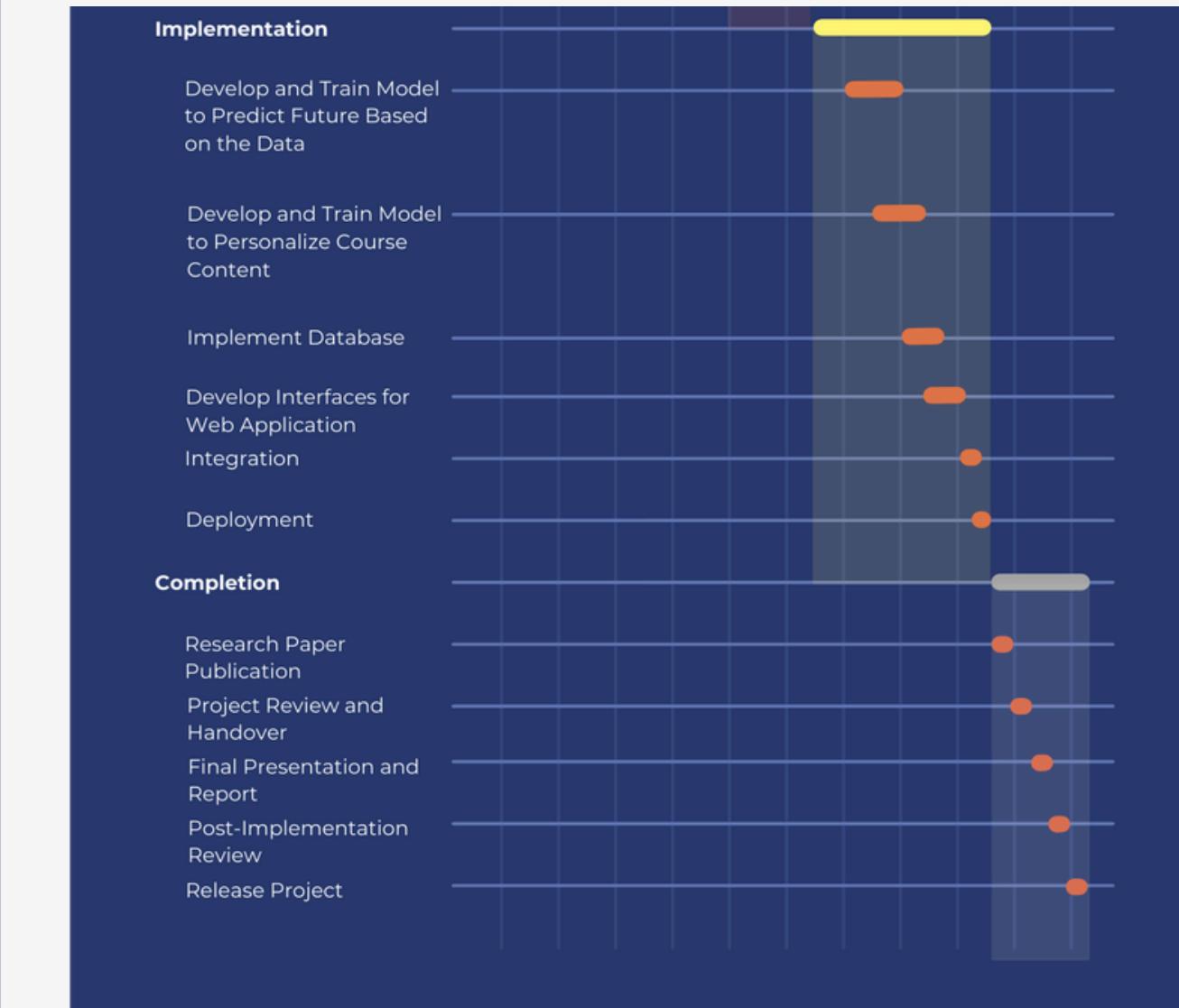
- Data Collection -100% ( before January 3rd week)
- Future Prediction model and Dynamic difficulty leveling Model and designs



## WORK BREAKDOWN STRUCTURE



# GANNT CHART



# REFERENCES

- . Akçapınar, G., Hasnine, M. N., Majumdar, R., Flanagan, B., Ogata, H. (2019). Developing an earlywarning system for spotting at-risk students by using eBook interaction logs. *Smart Learning Environments*, 6, 1-15. <https://doi.org/10.1186/s40561-019-0083-4>
- Adnan, M., Habib, A., Ashraf, J., Mussadiq, S., Raza, A. A., Abid, M., Bashir, M., Khan, S. U. (2021). Predicting at-risk students at different percentages of course length for early intervention using machine learning models. *IEEE Access*, 9, 7519-7539.
- Nimy, E., Mosia, M., Chibaya, C. (2023). Identifying At-Risk Students for Early Intervention—A Probabilistic Machine Learning Approach. *Applied Sciences*, 13, 3869. <https://doi.org/10.3390/app13063869>
- Er, E. (2012). Identifying at-risk students using machine learning techniques: A case study with IS 100. *International Journal of Machine Learning and Computing*, 2, 476. <https://doi.org/10.7763/IJMLC.2012.V2.171>



# AI - STUDENT ASSISTENT

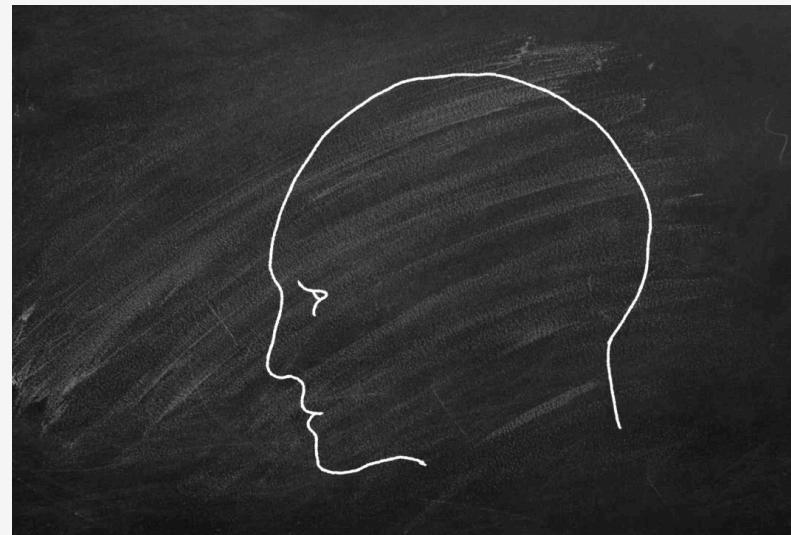
---



# BACKGROUND

- Focus on enhancing listening and comprehension skills.
- Interactive listening activities with recorded conversations.
- Students answer questions related to the audio.
- Automatic evaluation of student responses for accuracy.
- Encourages critical thinking and language development.

# RESEARCH PROBLEMS



- Accuracy of Speech Recognition in Evaluating Student Responses
- Impact of Student Engagement and Learning Outcomes
- Effectiveness of Audio Content in Enhancing Listening and Comprehension Skills
- Adapting Listening Activities to Different Learning Paces and Styles



# SUB OBJECTIVES

01

Question Collection for Main  
Listening Activities and Practice  
Sessions

02

Machine Learning Model for Answer  
Similarity Check for Listening Activity

03

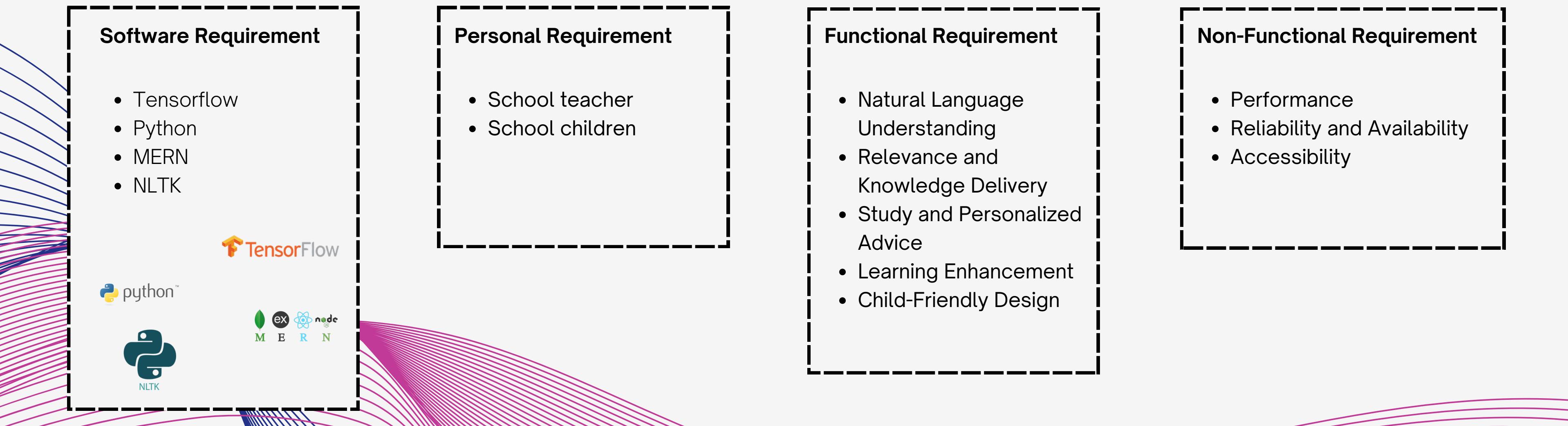
Create a Model for Check Student's  
Pronounsiation

04

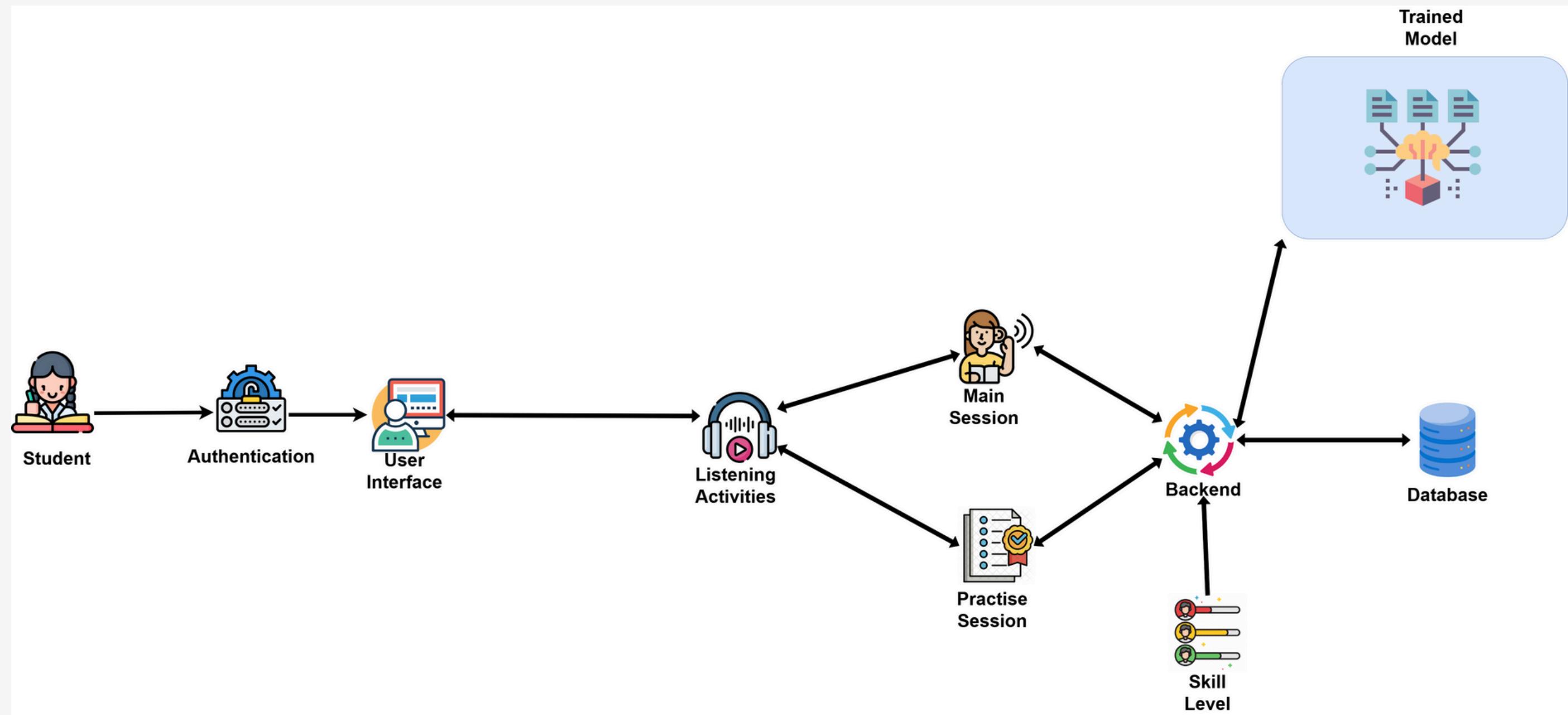
Create a child-friendly interface to  
enhance learning accessibility and  
interest

# METHODOLOGY

## SYSTEM, PERSONAL, AND SOFTWARE REQUIREMENT SPECIFICATION



# System Diagram





# Data Collection and Annotation

## Data Collection Methods

- From the students at the Regent Language Institute, Negombo
- From the External Supervisor

## Annotation

- From the External Supervisor





## Proof of work

```
Test Predictions: [0.95188403 0.82827616]  
Test Loss (MSE): 0.05522859410356091  
Test MAE: 0.19008009433746337  
Test MSE: 0.05522859410356091  
Test R2: 0.9999999999999998  
PS C:\SLIIT\4th Year\Research\similarity_model> █
```

```
Test Ground Truth: [0.85 0.7 0.78 0.65 0.8 ]  
Test Loss (MSE): 0.037866019860783855  
Test MAE: 0.17525903606414794  
Test MSE: 0.037866019860783855  
Test R2: 0.07156103269282593  
Test Accuracy: 20.00%  
PS C:\SLIIT\4th Year\Research\similarity_model> █
```



## Proof of work

```
Test Predictions: [0.95188403 0.82827616]
Test Ground Truth: [0.9 0.5]
Test Loss (MSE): 0.05522859410356091
Test MAE: 0.19008009433746337
Test MSE: 0.05522859410356091
Test R2: 0.9999999999999998
Test Accuracy (within ±0.1): 50.00%
PS C:\SLIIT\4th Year\Research\similarity_model>
```

```
55/55 [=====] - 840s 14s/step - loss: 0.1215 - mae: 0.3025
Epoch 2/5
55/55 [=====] - 819s 15s/step - loss: 0.1015 - mae: 0.2761
Epoch 3/5
55/55 [=====] - 822s 15s/step - loss: 0.0849 - mae: 0.2488
Epoch 4/5
55/55 [=====] - 839s 15s/step - loss: 0.0725 - mae: 0.2285
Epoch 5/5
55/55 [=====] - 831s 15s/step - loss: 0.0637 - mae: 0.2123
Model saved as 'similarity_model'
1/1 [=====] - 9s 9s/step
Test Predictions: [0.99926794 0.6062396 ]
Test Loss (MSE): 0.010570489630753525
Test MAE: 0.10275377631187438
Test MSE: 0.010570489630753525
Test R2: 0.9999999999999996
```



66

## COMPLETED COMPONENTS

- Data Collection - 50%
- ML Model for get answer correction for paper

66

## FUTURE DEVELOPMENT

- Data Collection -100% ( before January 3rd week)
- Create a Model for Check Student's Pronounsiation



## WORK BREAKDOWN STRUCTURE



# GANNT CHART



# REFERENCES

- [1] "Learning Spanish with Duolingo: How Well Does It Work?" - 2023 Conference on Language Learning Technologies, August 15-19, 2023, Barcelona, Spain.
- [2] "The Implementation of ELLLO Web-Based Application in Teaching Listening at MA Al-Ikhlas Ujung Bone" - 2024 International Conference on Education Technology, March 10-13, 2024, Jakarta, Indonesia
- [2] "The Implementation of ELLLO Web-Based Application in Teaching Listening at MA Al-Ikhlas Ujung Bone" - 2024 International Conference on Education Technology, March 10-13, 2024, Jakarta, Indonesia
- [3] "The Role of BBC Learning English Podcasts in Developing Speaking Skills Among EFL Learners" - 2023 International Conference on Language and Communication, September 20-23, 2023, London, UK.
- [5] "Enhancing Listening Skills and Learning Specific Language with Transcription Activities Using Lyrics Training" - 2024 International Conference on Language Learning Technologies, June 15-18, 2024, Madrid, Spain.



# AI-BASED INTERACTIVE VOICE ASSESSMENT

---



# BACKGROUND

- Developing strong language skills including pronunciation and fluency is crucial for children aged 10-12, as this is a critical period for their cognitive and social development.
- Traditional language learning methods, such as classroom instruction and rote memorization, often fail to engage students fully and address their individual needs.
- To address these issues, we are going to use advanced AI-powered technologies to design an interactive and personalized learning environment.
- These technologies will provide quick feedback, allowing for real-time practice and targeted assistance.
- We aim to increase student engagement and empower them to develop their language abilities through the use of voice recognition and natural language processing, enabling improved communication for academic and social success.

# REASEARCH GAP

Project 1 - Developing an AI-Assisted Low-Resource Spoken Language Learning App for Children.

Project 2 - The effectiveness of computer-assisted pronunciation training for foreign language learning by children

Project 3 - Duolingo Language Learning Platform (existing System)

Study	Immediate Pronunciation Feedback	Pre-Assessment Pronunciation Assistance	Contextual Vocabulary Enhancement	Alignment with School Syllabus
Project 1	X	X	X	X
Project 2	X	✓	X	X
Project 3	✓	X	X	X
Proposed System	✓	✓	✓	✓

# RESEARCH PROBLEMS



**How can real-time feedback on pronunciation improve the speaking skills of children aged 10-12?**

- When children receive instant feedback, they can quickly identify and correct their mistakes

**What impact does pre-assessment pronunciation assistance have on students' reading and speaking performance?**

- Pre-assessment pronunciation assistance allows students to listen to correct pronunciations before attempting to read aloud

**What are the benefits of providing contextual vocabulary enhancement through an interactive word definition system?**

- Providing contextual vocabulary enhancement through an interactive word definition system helps students understand and retain new words more effectively

**How does tracking individual performance over time with visual progress insights aid in identifying areas for improvement and enhancing learning outcomes?**

- Tracking individual performance over time and providing visual progress insights helps in identifying specific areas where students need improvement, allowing for targeted interventions.



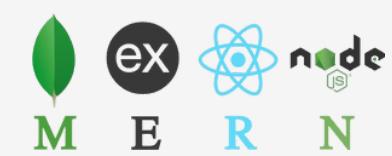
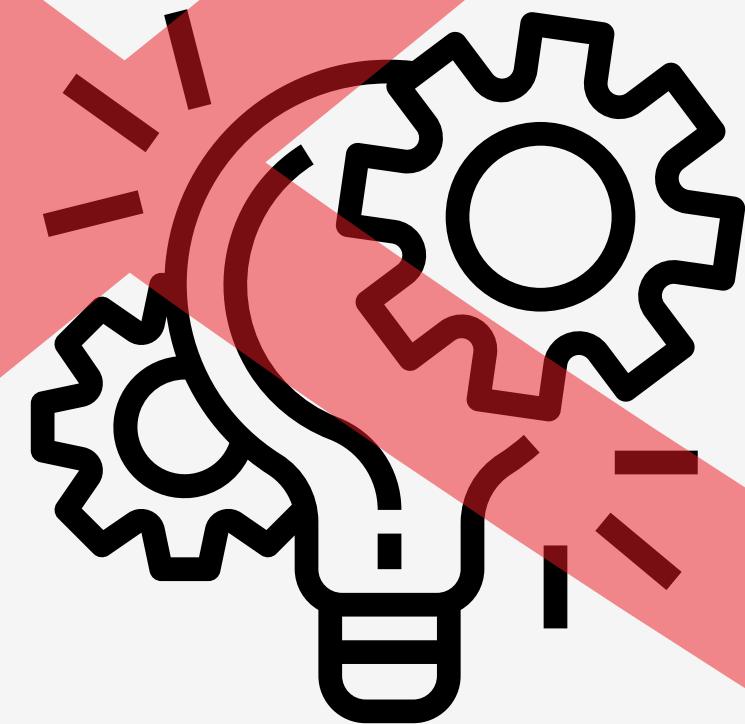
# SUB OBJECTIVES

- 01 Implement a functionality to Capture and record student pronunciations
- 02 Implement a functionality to convert student audio to text during reading sessions.
- 03 Identify pronunciation errors by comparing transcribed text with the original text.
- 04 Develop algorithms to provide instant feedback on pronunciation errors.
- 05 Implement functionality to convert text into audio for students to hear correct pronunciations.
- 06 Implement functionality to provide definitions and audio pronunciations of selected words.

# METHODOLOGY

## TECHNOLOGIES

MERN Stack  
Python  
Tensorflow  
WebRTC  
Google Cloud  
Git  
Merriam-Webster API



# SYSTEM, PERSONAL, AND SOFTWARE REQUIREMENT SPECIFICATION

## Software Requirement

- MERN Stack
- Tensorflow
- Python
- WebRTC
- Merriam-Webster API

## Personal Requirement

- School teacher
- School children

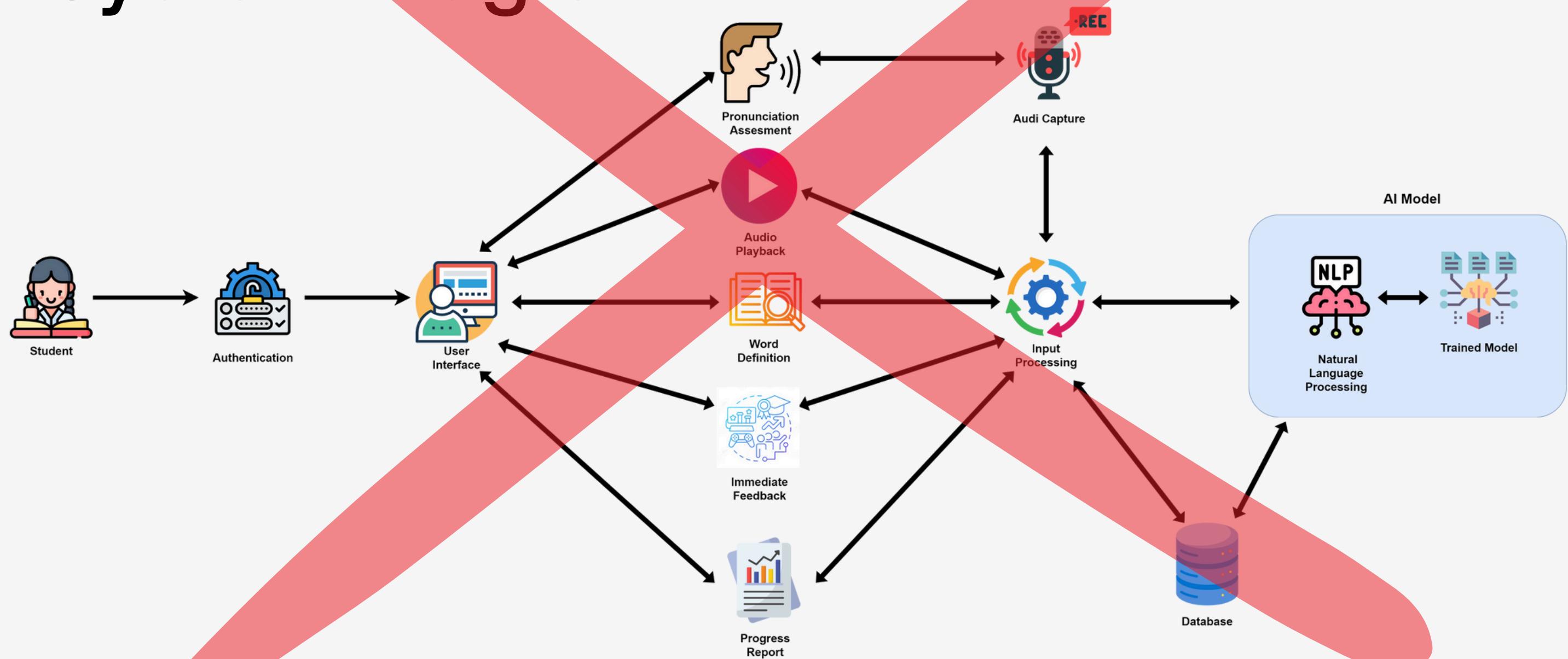
## Functional Requirement

- User Authentication
- Data Collection
- Reading Material Presentation
- Speech-to-Text Conversion
- Pronunciation Feedback
- Pre-Assessment Pronunciation Assistance
- Contextual Vocabulary Enhancement
- Progress Tracking

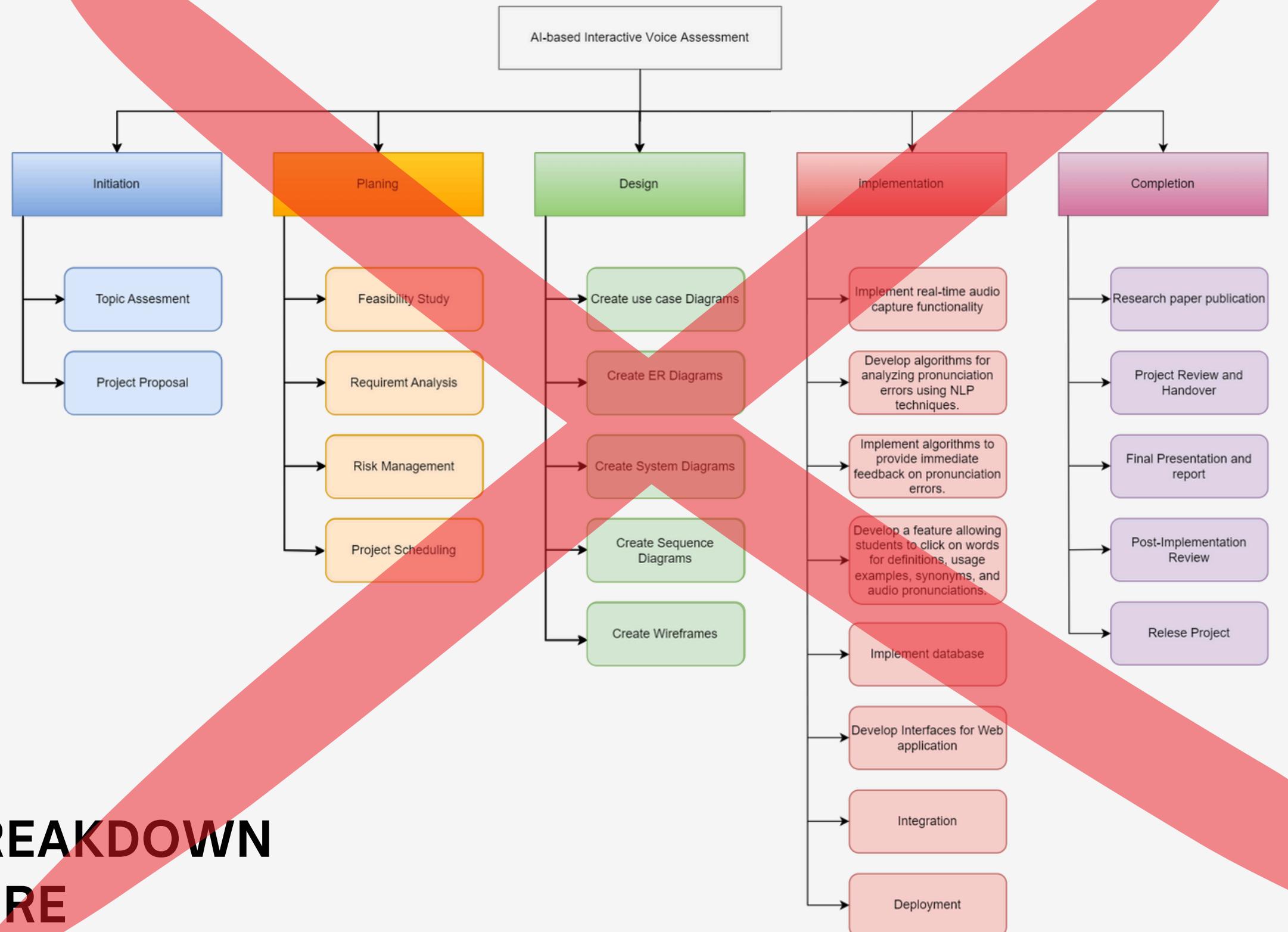
## Non-Functional Requirement

- Performance
- Reliability and Availability
- Security
- The system must be able to handle a large number of requests

# System Diagram

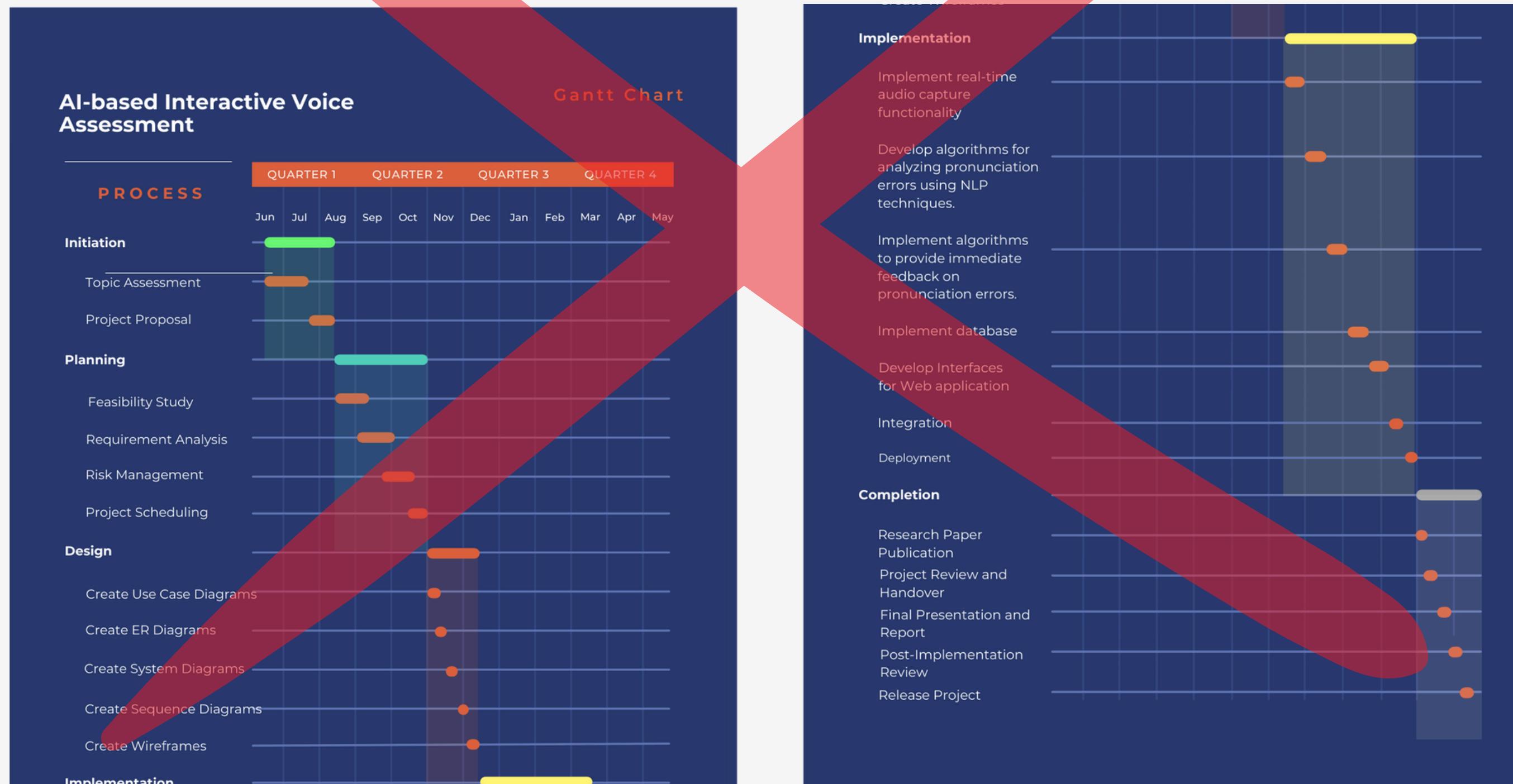


## WORK BREAKDOWN STRUCTURE





# GANNT CHART



# REFERENCES

- J. Xu, H. Wang, and Z. Li, "Developing an AI-Assisted Low-Resource Spoken Language Learning App for Children," Proceedings of the 2023 Conference on Human Factors in Computing Systems, 2023.  
<https://ieeexplore.ieee.org/document/10214276>
- The effectiveness of computer-assisted pronunciation training for foreign language learning by children.  
<https://www.tandfonline.com/doi/abs/10.1080/09588220802447651>
- Duolingo, "Duolingo Language Learning Platform," 2024.  
<https://www.duolingo.com/>

# THANK YOU