**Faculty Kit**

The faculty kit contains the evaluation strategy for the different milestones of the project and any other documents/links that may aid in the evaluation process (like sample quizzes on technologies etc)

**Faculty Kit: Emotion-Based Music Recommender System**

**1. Requirements Specification Evaluation**

**Objective**: Ensure students understand and clearly define the system requirements for each project component.

* **Clarity**: Check if the requirements for emotion detection, music recommendation, and UI are clearly stated.
* **Feasibility**: Are the assumptions valid and justifiable in the context of the project?
* **Team Understanding**: Each team member should fully understand their assigned tasks (e.g., emotion detection, Spotify API integration, UI development).
* **Presentation Quality**: Evaluate how well the team presents the system's core requirements and objectives.

**2. Technology Familiarization**

**Objective**: Assess the students' grasp of the technologies they will use (Python, OpenCV, TensorFlow/PyTorch, Spotify API).

* **Presentation**: Each member presents the technologies related to their responsibility (e.g., emotion detection models, Spotify API, UI frameworks).
* **Quiz**: Faculty can test the team's knowledge on these tools and technologies via a quiz, covering areas like machine learning models, API integration, and UI development.

**3. Database Design Evaluation**

**Objective**: Evaluate the database that supports user information, emotion records, and music recommendations.

* **Clarity & Redundancy**: Ensure that database fields for storing user data, emotions, and music recommendations are well-defined, and check for redundancy.
* **Backup & Recovery**: Verify if students have implemented or planned a backup and recovery system for data.

**4. High-Level and Detailed Design Evaluation**

**Objective**: Assess the design documents and presentations for each module.

* **Requirements Coverage**: Check if the design covers all specified requirements (e.g., emotion detection, Spotify integration, user interface).
* **Pseudocode & Flowcharts**: Evaluate the provided pseudocode or flowcharts for detail and correctness (e.g., for emotion detection, music recommendation, and UI flow).
* **Error Handling & Alternatives**: The design should anticipate potential errors (e.g., webcam issues, API failures) and propose alternative solutions.

**5. Front-End Implementation**

**Objective**: Evaluate the user interface (UI) created for interacting with the emotion detection and music recommendation modules.

* **UI/UX**: Is the interface intuitive and easy to navigate? Assess the balance between functionality and aesthetics.
* **Error Handling**: Does the UI provide meaningful error messages for incorrect inputs or system failures?
* **Help Documentation**: Check if the help screens match the actual UI and are easy to understand for users.

**6. Integration of Front-End with Backend and Spotify API**

**Objective**: Test how well the front-end integrates with the backend and Spotify API.

* **Full Demo**: The students should demonstrate the system, showing seamless interaction between the frontend (UI), backend (emotion detection), and Spotify API.
* **Stability & Robustness**: Assess the stability of the application when executing core functionalities, such as emotion detection and music recommendation.
* **Major Feature Demonstration**: Ensure that all significant features (e.g., user login, emotion detection, music playback) work smoothly without system crashes or performance issues.

**7. Test-Plan Review**

**Objective**: Ensure that the system is tested for all edge cases and that the test plan covers all requirements.

* **Coverage**: Evaluate if the test cases cover all key functionalities mentioned in the Student Kit (emotion detection, music recommendation, UI navigation).
* **Clarity & Completeness**: Ensure that the test cases are clearly written, with descriptions of how to execute them, and cover both normal and error scenarios (e.g., multiple facial expressions, API failures).
* **Exception Handling**: Verify if the system can handle unexpected scenarios like webcam failures or Spotify API issues.

**8. Final Demo & Review**

**Objective**: The final stage involves evaluating the overall functionality and performance of the system.

* **Demo**: The team should perform a final demo of the system, covering emotion detection, music recommendation, and user interaction.
* **Report**: Evaluate the final project report to ensure it documents the entire project lifecycle, including design decisions, technologies used, and challenges faced.
* **Intermediate Submissions**: Consider presentations, code samples, and write-ups submitted throughout the project as part of the final evaluation.

**9. Project Evaluation Rubrics:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | **Sub-Category/Factors** | **Marks** | **Criteria for Marks Allocation** |
| 1. Requirements Specification | Clarity of Requirements | 5 | Clear, concise, and feasible requirements earn full marks. Partial clarity loses marks. |
|  | Validity of Assumptions | 5 | Well-justified and realistic assumptions score higher. |
|  | Understanding by Team Members | 5 | All members demonstrate equal understanding of the project. |
|  | Presentation Quality | 5 | Effective and professional presentation, clear slides, structured delivery. |
| 2. Technology Familiarization | Knowledge of Technologies | 5 | All members must demonstrate understanding of chosen tools (Python, OpenCV, TensorFlow/PyTorch, Spotify API). |
|  | Presentation of Technology | 5 | Clear and informative presentations on each technology component. |
|  | Quiz on Technology | 5 | Accurate answers in the technology quiz reflecting deep understanding. |
| 3. Database Design | Clarity of Database Fields | 5 | All fields well-defined, with no redundancies. |
|  | Backup and Recovery Plan | 5 | A strong plan for data backup and recovery implemented. |
|  | Efficiency of Design | 5 | Efficient data storage and database structure without unnecessary complexity. |
| 4. High-Level and Detailed Design | Requirement Coverage | 5 | Design covers all requirements from the specification document. |
|  | Pseudocode/Flowcharts Details | 5 | Detailed and understandable pseudocode/flowcharts that match the implementation. |
|  | Error Handling and Alternatives | 5 | Robust handling of edge cases and alternative designs considered. |
| 5. Front-End Implementation | UI Design (Look and Feel) | 5 | Intuitive, visually appealing UI design that enhances usability. |
|  | Error Messaging | 5 | Clear and meaningful error messages throughout the system. |
|  | User Help Documentation | 5 | User guides/help documentation are clear, concise, and match implementation. |
| 6. Integration with Backend | Seamless Front-End to Back-End Integration | 5 | Proper interaction between the UI, emotion detection, and Spotify API without issues. |
|  | Stability of Application | 5 | System runs without crashing, shows stability during demonstration. |
|  | Feature Completion | 5 | Major features (emotion detection, music recommendation, etc.) are fully functional. |
| 7. Test Plan Review | Test Case Completeness | 5 | All major requirements are covered in test cases. |
|  | Execution of Test Cases | 5 | Clear and correct execution of test cases during the evaluation. |
|  | Handling of Exceptions and Errors | 5 | Exception scenarios are fully tested and handled gracefully. |
| 8. Final Demo & Project Report | Final Demo Quality | 5 | Comprehensive demo covering all features with clarity and professionalism. |
|  | Final Project Report | 5 | Well-documented project report, detailing the entire process from design to implementation. |
|  | Intermediate Submissions | 5 | Consistent quality in all intermediate presentations and submissions. |

**Additional Resources:**

* Python and OpenCV tutorials for image processing
* TensorFlow/PyTorch documentation for emotion detection models
* Spotify API documentation and example implementations
* UI/UX design principles and best practices

**(Signature):**

**Faculty Name: Praveen Kumar Yadav**