**Project Design Phase-II**
**Technology Stack (Architecture & Stack)**

| | |
|---|---|
| Date | 23 October2022 |
| Team ID | PNT2022TMID40383 |
| Project Name | Project – Emerging Methods for Early Detection of Forest Fires |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

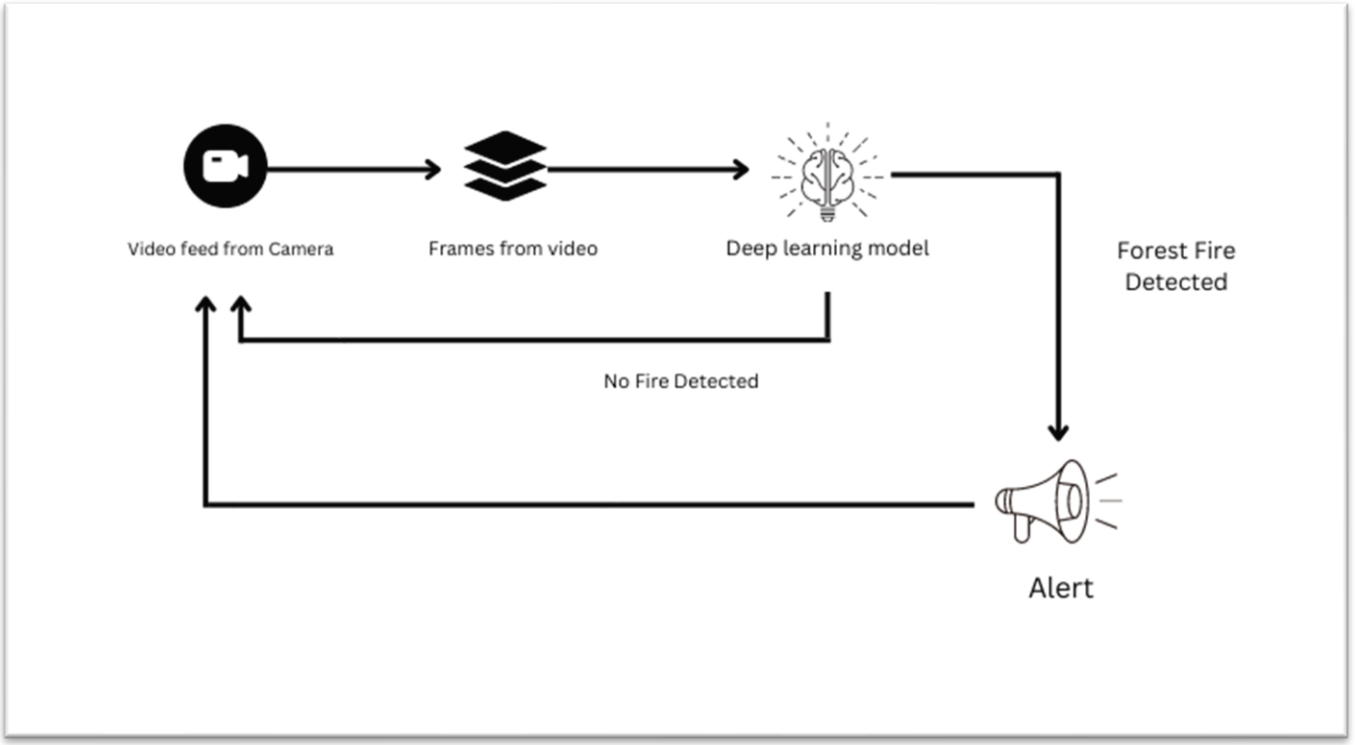

**Table-1: Components & Technologies:**

| S. No | Component | Description | Technology |
|---|---|---|---|
| 1. | User Interface | Website or Web app | HTML, CSS, JavaScript / React Js. |
| 2. | Application Logic-1 | Model building and training | Python |
| 3. | Application Logic-2 | Getting image or video data from user from prediction | IBM Watson STT service |
| 4. | Application Logic-3 | Fetch the relevant data from the database and display them to user | IBM Watson Assistant |
| 5. | Database | Image and video data captured at the forest | MySQL, NoSQL |
| 6. | Cloud Database | Fetch data from database and feed them to model for prediction | IBM DB2, IBM Cloudant etc. |
| 7. | File Storage | Video data, Image data, Login credentials and API keys | IBM Block Storage |
| 8. | External API-1 | To get data from the database when user give the video input | IBM Storage API |
| 9. | External API-2 | To get the username and password of the specific user | Authentication API |
| 10. | Machine Learning Model | To predict the cause of the forest fire and its exact location | Image Recognition Model |
| 11. | Infrastructure (Server / Cloud) | To deploy application in cloud server | Cloud Foundry |

**Table-2: Application Characteristics:**

| S. No | Characteristics | Description | Technology |
|---|---|---|---|
| 1. | Open-Source Frameworks | Application is built by using flask | WSGI framework (Web Server Gateway Interface) |
| 2. | Security Implementations | For authenticating the user data and protecting the data about causes of fire at the forest in the database | SHA-256 / Encryption / OWASP / IAM Controls |
| 3. | Scalable Architecture | To scale our application in server side by supporting users including web browsers, mobile browsers etc., | IBM Auto Scaling |
| 4. | Availability | To make application available both online and offline and 24/7 service | IBM Cloud load balancer |
| 5. | Performance | Designing an application which provides accuracy in prediction | IBM instance |