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

| | |
|---|---|
| Date | 08 October 2022 |
| Team ID | PNT2022TMID40761 |
| Project Name | **NATURAL DISASTERS INTENSITY ANALYSIS AND CLASSIFICATION USING ARTIFICIAL INTELLIGENCE** |

**Technical Architecture:**

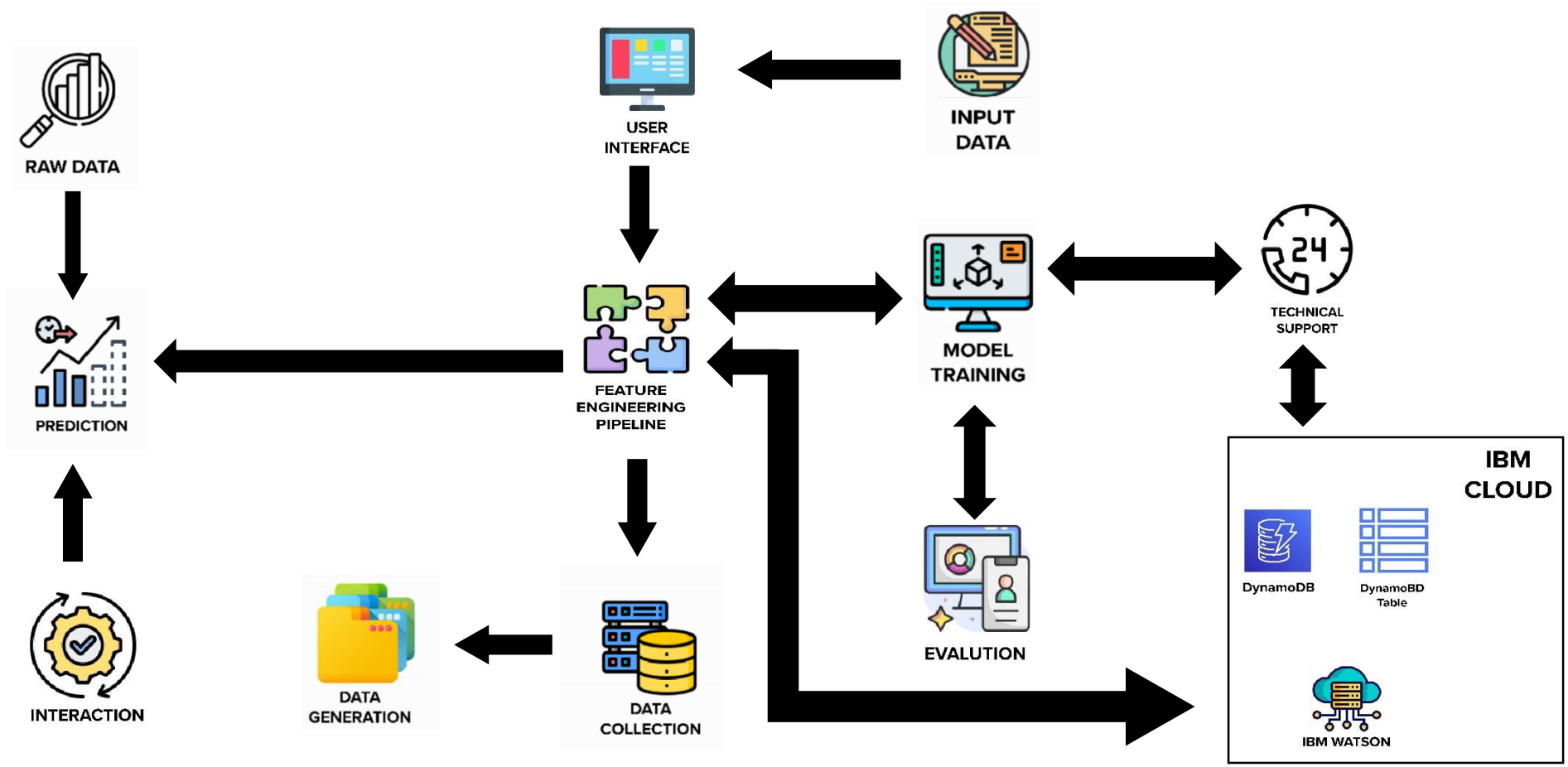

**Table-1: Components & Technologies:**

| S. No | Component | Description | Technology |
|---|---|---|---|
| 1. | User Interface | User interacts with application for the prediction of Any Natural disaster which will happen in future minutes. | HTML, CSS, JavaScript, Django, Python. |
| 2. | Feature Engineering Pipeline | Algorithms can't make sense of raw data. We have to select, transform, combine, and otherwise prepare our data so the algorithm can find useful patterns. | Image processing, pattern extraction, etc. |
| 3. | Model Training kit | It learns patterns from the data. Then they use these patterns to perform particular tasks. | Multiclass Classification Model, Regression Model, etc. |
| 4. | Prediction unit | This function is used to predict outcomes from the new trained data to perform new tasks and solve new problems. | Decision trees, Regression, Neural networks. |
| 5. | Evaluation system | It monitors that how Algorithm performs on data as well as during training. | Chi-Square, Confusion Matrix, etc. |
| 6. | Interactive services | To interact with our model and give it problems to solve. Usually this takes the form of an API, a user interface, or a command-line interface. | Application programming interface, etc. |
| 7. | Data collection unit | Data is only useful if it’s accessible, so it needs to be stored ideally in a consistent structure and conveniently in one place. | IBM Cloud, SQL Server. |
| 8. | Data generation system | Every machine learning application lives off data. That data has to come from somewhere. Usually, it’s generated by one of your core business functions. | Synthetic data generation. |

| | | | |
|---|---|---|---|
| 9. | Database management system | An organized collection of data stored in database, so that it can be easily accessed and managed. | MySQL, DynamoDB etc. |
| 10. | IBM Cloud services | Processed data stored in cloud service which can be access by the admin anywhere over the internet. | IBM Cloud etc. |

**Table-2: Application Characteristics:**

| S. No | Characteristics | Description | Technology |
|---|---|---|---|
| 1. | Open-Source Frameworks | An open source framework is a template for software development that is designed by a social network of software developers. These frameworks are free for public use and provide the foundation for building a software application. | Keras, pensor flow. |
| 2. | Authentication | This keeps our models secure and makes sure only those who have permission can use them. | Encryption and Decryption (OTP). |
| 3. | Application interface | User uses mobile application and web application to interact with model | Android and Web Development (PhoneGap, ReactNative, and NativeScript). |
| 4. | Availability (both Online and Offline work) | Its include both online and offline work. As good internet connection is need for online work to explore the software perfectly. Offline work includes the saved data to explore for later time. | Caching, backend server. |
| 5. | Regular Updates | The truly excellent software product needs a continuous process of improvements and updates. Maintain your server and make sure that your content is always up-to-date. Regularly update an app and enrich it with new features. | • Waterfall Approach<br>• Incremental Approach<br>• Spiral Approach |
| 6. | Personalization | Software has features like flexible fonts, | |

| | | | |
|---|---|---|---|
| | | backgrounds, settings, colour themes, etc. which make a software interface looks good and functional. | • HubSpot<br>• Proof |