Women's Maltreatment Redressal System based on Machine Learning Techniques

Hemanth adhi sekhar obulasetty, Preetha Sivakumar, Shalini L

Student, Senior Assistant Professor, Vellore Institute of Technology, Vellore (Tamil Nadu), INDIA

Abstract - *A significant and expanding concern on a global scale is violence against women. Various issues like difference in maintenance, proper maintenance of records in different countries, even parts of a country are present in the current system. Through our project, we shall be addressing these issues by development of a portal to register and assign complaints according to the classification based on ML program. The main focus on the portal would be ease the communication between the involved victim and government officials. In addition to which, new features like real –time communication through WEBRTC protocol, victim services and SOS alert are included. We have extensively analyzed and classified the case based on the selected algorithm having highest accuracy namely deep feed forward algorithm. The implemented algorithm is the simplest form of artificial neural network having no loops and carries the data only in a single direction. Known as the multiple layer perceptron, it inputs enter the layer and are multiplied by the weights in this model comprising of multiple hidden layers which are summed together to form a total. With the accurate and immediate classification of the cases, the efficiency and execution will be immensely improved through the use of our application.*

Key Words: **Web application, Deep Feed Forward, ML Algorithms, WebRTC, Deep learning**

1. INTRODUCTION

Misuse or violent behaviour is as old as humanity, but it takes many forms and levels as time passes. Women's violence, in particular, is a major issue that must be addressed. We discovered that there is no link between cases involving the same person in different regions, and that following up on details and the overall process is difficult for police authorities due to the lack of interconnection between different regions and communication between involved officials. People of different cultures, background and education levels even to illiteracy levels require some mode to register their complaint. With several cases being filed every day, knowing the priorities of the cases is essential.
Assigning the importance of the case was employed using machine learning algorithms from classical approaches like K-nearest neighbours, support vector machine, decision trees, random forest and gradient boosting algorithms to deep feedforward neural network, a deep learning method. The model was intensively trained after data processing, cleaning and feature selection to give the most accurate prediction. A centralized platform with all related information and services, as well as a forum for users, administrators, and government officials, is required. To address this issue, we developed our project.

1.1 Methodology:

- Web applications for complaint register and other services
- Deep learning based classification of case records which is to assigned to the government officers

2. Overview of Application

The project intends to create a universal portal as a friendly and convenient space for the victims to communicate and register their complaint and experiences officially. Upon proper authentication and sign up, the users would gain access to the resources like blood bank information, laws and policies up to date and global statistics in addition to SOS alert, tracking case and real-time communication through WebRTC. The main feature, registration, involves filing a form consisting of name of accuses, identification marks, details of incident and its location which can be either verbal or through video call. Depending on the severity of cases, it will be immediately assigned to a police officer of the concerned rank. All information stored (using Mysql) and maintained is secured and private to ensure no data is erroneously used and exploited.

Government officials have a separate login gateway and with authentication would be given access to the homepage containing all case records. They also have the option to view victim's details and laws and order information and export database of complaint records as a pdf. Their history of cases can also be seen inclusive of complaint, status and solved by. Video call option is given to communicate to those that feel more comfortable or unable to fill the digital form to file an official complaint.

The admin gains to responsibility to store and maintain the database and portal with accurate information including the static records of global statistics, law and policies and blood bank. The admin insures the complaints are being accurately classified and solved immediately. Contact forms which may be include feedback, issues or doubts regarding the application is collected and performs the necessary steps in regards to the user's message.

2. Relevant Concepts

2.1 WebRTC

Web Real-Time Communication, is a developing standard for real-time browser communication. According to many IT experts, WebRTC will eventually lead to a breakthrough in communication technology. Because users do not need to install plugins such as Adobe Flash or use third-party software such as Skype, WebRTC's no-plugins strategy is advantageous and significantly reduces setup time. It enables real-time voice, video, and data transmission capabilities via web browsers. WebRTC is well-known for its exceptional peer-to-peer communication features such as interoperability, security, and video quality. However, it is not without flaws. The difficulties encountered when using WebRTC are caused by the diversity of access methods, as their capacities and networks differ. The application is affected by the user's network bandwidth and latency. Screen size is a factor because resolutions and quality vary, making it impossible to broadcast equal quality to all users.

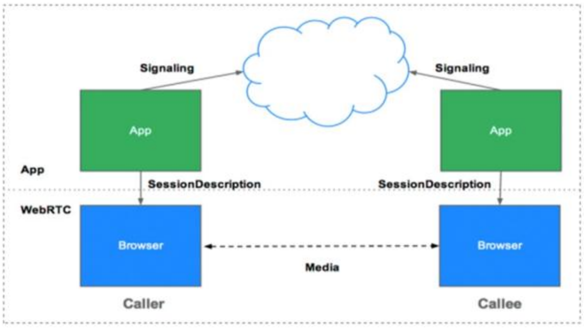


Fig -1: WebRTC Architecture

2.2 Deep learning

Machine learning focuses on the creation of algorithms and mathematical analysis that enables computers to learn and make predictions or judgements without being explicitly programmed. It entails training algorithms on big datasets to detect patterns and relationships, and then using these patterns to forecast or make choices about incoming data. Deep learning is fragment of machine learning that evaluates complicated patterns and correlations in data using neural networks with numerous layers. Like the complex structure and functionality of the human brain, the deep learning algorithms are trained to compute broad spectrum of tasks including as image identification, natural language processing, and speech recognition. The major limitation includes data dependency, overfitting, lack of interpretability, and computationally demanding.

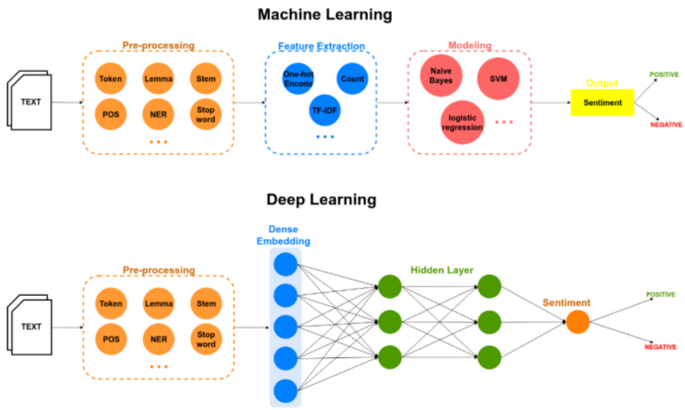


Fig -2: Difference between ML and Deep Learning

Out of the numerous deep learning architectures, the most popularly used are feedforward neural networks, convolutional neural networks and recurrent neural network.

The deep learning technique utilized will fluctuate depending on the kind of application and dataset used. The general workflow starts with an in-depth grasp of the problem specification and its viability. Following that, an appropriate dataset is selected, and the suitable algorithm type is chosen. Going to follow that, the data is pre-processed and cleaned to include the appropriate features and data format for training. The model's performance is evaluated after achieving an optimal accuracy score by hyper-tuning

2.3 Data Visualization

Data visualization is the concept of converting information into a visual context via captions, maps, or graphs in a manner that allows the brain to easily comprehend and analyze data. It facilitates the process of identifying patterns, trends, and outliers in large data sets by using a dashboard that contains informatics graphics, visuals, and statistics. It is widely used in the data science process after data is collected, processed, and modelled. It is capable of identifying, locating, manipulating, formatting, and delivering data in the most efficient manner possible. Users may diversify from educators using it to display student test results to computer scientists for advance artificial intelligence (AI) analyzes, and executives can use it to share information with stakeholders. The prominent software used for visualization tableau which is used for our project.

2.4 Gradient Boosting

Boosting is an ensemble modelling technique that creates a strong classifier from a set of weak classifiers. A model from training data is corrected after errors, and models are added until either the entire training data set is correctly predicted or the maximum number of models are added. Gradient boosting was taken a step further, with each predictor correcting the error of its predecessor. It is mainly used in classification and regression problems. Given its practicality in dealing with missing data, outliers, and large cardinality categorical values on your features even without adjuvant, this technique is frequently utilised in many applications. The kind of input (numeric and categorical), managing missing information, and flexibility are all advantages. Overfitting, on the other hand, can occur and is computationally costly. The 4 types of gradient boosting includes gradient boosting machine, xgboost, catboost and lightgbm. All of which provide high accuracy, but is chosen depending on the application and data chosen given.

3. Dataset Features

Our application is based on a Kaggle dataset of violent crimes against women in various countries comprising of training and testing data. In total, the data consists of 13638 records and 329 features which are inconsistent with textual and numerical data and has unprocessed information. The training data has grievance description along with the rank where 1 indicating low urgency and 4 implicating immediate attention. Data features were filtered and selected based on Extra Tree classifier to remove redundant material thus increasing the accuracy.

Feature	Description
Issues	Description of distinct issues present in dataset
Paragraphs_41, Paragraphs_34, Paragraphs_35, Paragraphs_35-1, Paragraphs_8-2, Paragraphs_29-3, Paragraphs_10-2, Paragraphs_6-1, Paragraphs_8-1	Law id and description linked with a type of issue.
Complaint_id	Unique id of complaint
Rank	Level of importance of issue, ranging from 1- 4
Article_35, Article_41, ccl_article_6, Article_29	Name of law article applicable to the case
Share_point_id	Encoded values
Incident_location	Location were incident took place
Res_country	Name of country responding to the complaint incident
Separate_opinion	Binary values representing valid and invalid
Document_id_c	The document is to respective of category chamber
Document_id_gc	The document is to respective of category grand chamber
Category	Type of issue
Document_id_comm	The document is to respective of category committee
Applicability	Types of relevancy linked with case
Type_description	Different types of issue present
Interval_intro_decision	Time taken to give decision from complaint registered date
Interval_intro_judgement	Time taken to pass judgement from complaint registered date
Interval_decision_judgement	Time take to pass judgement from decision date

4. Data Pre-processing

Data pre-processing is a technique for transforming data into a more accessible and efficient format. In real-time, most raw data contains null values, superfluous values, duplicate values, and noise with no order or trend, which can have a significant impact on the model's performance. This pre-processing approach cleans, consistencies, and organises the data in order to adequately train the dataset for an efficient model. There are numerous methods for initialising this technique, including filling NA or NULL values, deleting duplicate values, dealing with outliers, normalising the data to make it scale free, and smoothing to deal with noise, among others. Because our dataset is somewhat huge, we used a variety of data pre-processing techniques, including the following:

4.1 Amount of Missing Data

Missing_data = combined.isnull().sum() creates a new DataFrame missing_data that contains the count of

International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 10 Issue: 03 | Mar 2023 www.irjet.net p-ISSN: 2395-0072

missing values for each column of the train dataset. The isnull() function examines for missing values in the train dataset and returns missing data of the same structure with True for missing values and False otherwise. Additionally, sum() function is used to calculate the frequency of missing values in each column.

```
missing_data = combined.isnull().sum()
total_percentage = (missing_data.sum()/combined.shape[0])
print(f'The total percentage of missing data is {round(total_percentage,2)}%')
```

```
The total percentage of missing data is 33.27%
```

Fig -3: Missing Data Function

4. 2 Combining Issues

This function combines multiple columns containing issue information in a train dataset, the function first defines a list issue_columns containing the names of the columns that contain issue information. It results in a new DataFrame issue_df that contains only these columns from the original DataFrame df. The fillna() method is used to replace any missing values in the issue_df DataFrame with an empty string. The resulting string is assigned to the new 'issues' column. This code effectively combines all the issue columns in train dataset into a single column called 'issues'.

```
def combine_issues(df):
    issue_columns = [
        'issue.0', 'issue.1', 'issue.2', 'issue.3', 'issue.4', 'issue.5', 'issue.6', 'issue.7', 'issue.8',
        'issue.9', 'issue.10', 'issue.11', 'issue.12', 'issue.13', 'issue.14', 'issue.15', 'issue.16',
        'issue.17', 'issue.18', 'issue.19', 'issue.20', 'issue.21', 'issue.22', 'issue.23']
    issue_df = combined[issue_columns]
    issue_df.fillna('',inplace=True)
    issue_df['issues'] = issue_df[issue_columns].apply(lambda x: '. '.join([val for val in x if val != '']), axis=1)
    df.drop(issue_columns, axis=1, inplace=True)
    issue_df.drop(issue_columns, axis=1, inplace=True)
    df = pd.concat([df, issue_df], axis=1)
    return df
combined = combine_issues(combined)
```

Fig -4: Combing Issues Function

4.3 Discard Superfluous Features

Removes from a training dataset columns that only have a single unique value. These columns are deemed unnecessary since they provide no valuable information.

```
def remove_constant_values(df):
    print('Removing redundant columns -> ',)
    for col in df.columns:
        if df[col].nunique()==1:
            print(col,end=', ' )
            del df[col]
    return df
```

Fig -5: Discard Redundant Features Function

4.4 Featurize Columns

This function formulates new attributes by computing the absolute number of days between the combinatinons of 'decisiondate' and 'introductiondate', 'judgementdate' and 'introductiondate',and 'judgementdate' and 'decisiondate' using the dt.days attribute and the abs() method.

```
def featurize_date_columns(df):
df['daysbetween_intro_decision'] = ((pd.to_datetime(df['decisiondate']) - pd.to_datetime(df['introductiondate'])).dt.days).abs()
df['daysbetween_intro_judgement']=((pd.to_datetime(df['judgementdate']) - pd.to_datetime(df['introductiondate'])).dt.days).abs()
df['daysbetween_decision_judgement']=((pd.to_datetime(df['judgementdate']) - pd.to_datetime(df['decisiondate'])).dt.days).abs()
df.drop(['decisiondate','introductiondate','judgementdate'], axis=1, inplace=True)
return df
```

Fig -6: Featurize Columns Function

4.5 Selection of Suitable Features

Using the ExtraTreesClassifier, identifying the top 25 features with the largest feature importances and using the nlargest() method to plot them using a horizontal bar plot for better understanding.

```
from sklearn.ensemble import ExtraTreesClassifier
import matplotlib.pyplot as plt
model = ExtraTreesClassifier()
model.fit(X,y)
print(model.feature_importances_)
feat_importances = pd.Series(model.feature_importances_, index=X.columns)
feat_importances.nlargest(25).plot(kind='barh')
plt.show()
```

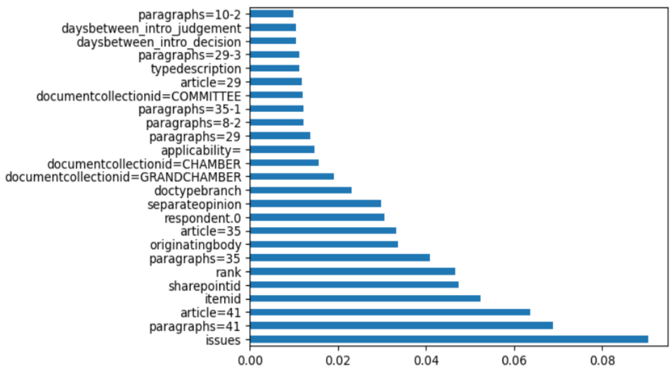


Fig -7: Selection of Features Function and Result

4. 6 Deletion of Features

This function deletes the other features columns apart from top 25, to train a simpler model with fewer features and more accuracy.

```
for col in combined_train.columns:
    if col not in ['issues','paragraphs=41','itemid','rank','article=41','sharepointid','originatingbody','respondent.0',
                   'article=35', 'paragraphs=35', 'separateopinion','documentcollectionid=CHAMBER',
                   'documentcollectionid=GRANDCHAMBER','doctypebranch','documentcollectionid=COMMITTEE',
                   'daysbetween_intro_decision','daysbetween_intro_judgement','daysbetween_decision_judgement',
                   'paragraphs=29','applicability=','paragraphs=35-1','paragraphs=8-2','typedescription',
                   'paragraphs=29-3','article=29','paragraphs=10-2','paragraphs=6-1','paragraphs=8-1','ccl_article=6',
                   'importance']:
        combined_train.drop(col,inplace=True,axis=1)
```

Fig -8: Feature Deletion Function

4.7 Features Scaling

This function transforms the training set using the transform() method of the StandardScaler object to standardize the features by subtracting the mean and dividing by the standard deviation. Feature scaling is a common pre-processing step in machine learning that helps to improve the performance of many model which is especially useful when working with features that have different scales or units. By scaling the features, we can ensure that they are all on a similar scale, which can help the model converge faster and make the results more interpretable.

```
X_scaler = StandardScaler().fit(X_train)
X_train_scaled = X_scaler.transform(X_train)
X_test_scaled = X_scaler.transform(X_test)
X_full_scaled = X_scaler.transform(X)
```

Fig -9: Feature Scaling Function

5. Machine Learning

The user's complaint serves as input to the machine learning system, which classifies the urgency of the case before assigning it to the appropriate police officer. The instances are prioritized and processed out instantly based on the output rank. The model utilized filters out unnecessary or duplicated forms, boosting the application's efficiency. The maintenance and arrangement of cases are methodical and effortless with suitable organization, resulting in further studies such as predominant case type, location of events, and social behavior. We may train and evaluate data for that model before deploying it. After making the necessary adjustments, the complaints will be saved in the database.

5.1 Classical ML Techniques

5.1.1 SVM

SVM is a prominent machine learning method that is used for classification and regression problems. SVM works by locating the hyperplane by data points are close as possible through which it divides the data into distinct groups. This approach can handle large datasets with many dimensions, efficiently working with tiny datasets unaffected by local optima. Some applications of SVM may be applied to binary and multi-class classification issues, forecasting the distance between the test point and the hyperplane in regression issues. SVM can be influenced by the kernel function used to translate data into a higher dimensional space. One drawback of SVM is that its computationally costly for big datasets and might take a long time to train.

5.1.2 KNN

K-Nearest Neighbours is a fundamental classification algorithm in Machine Learning, which comes under supervised learning algorithm that is widely used in pattern recognition, data mining, and intrusion detection. It is widely applicable in real-world scenarios, because it is non-parametric, which means it makes no underlying assumptions about data distribution. When dealing with large datasets, the robust, simple algorithm comes in handy, but it can be difficult to determine K value and has a high consumption cost. The approach is predicated on the notion that data points in the feature map that are close to each other belong to the same class. KNN is a lazy learner, which means it memorises the training data and utilises it at test time rather than learning a model from it. The time complexity and auxiliary space for the specified algorithm are O(N * logN) and O(N * logN), respectively O(1).

5.1.3 DECISION TREE

Decision trees, which recursively divide the data into subgroups based on the value of a feature, are another categorization approach. In order to optimise information gain or reduce impurity in the subsets, the partitions are chosen. Both qualitative and numerical data can be handled by the straightforward and clear DT algorithm. Even with limited datasets, DT performs well and is unaffected by outliers serving as its highlights. This approach is mainly used for classification problems (binary and multi-class) and regression by calculating target variable's mean or median for each leaf node. Factors like splitting criterion and the depth of the tree may influence its performance and result in overfitting if the tree is too deep or if the training data is noisy.

5.1.4 RANDOM FOREST

Random Forest is a commonly used algorithm for classification and regression problems using supervised learning. It is based on ensemble learning, which means that multiple classifiers assist in solving and enhance the model's performance. The model forecasts its final output using the majority votes of predictions from each tree, which would show a direct proportionality between the frequency of trees and accuracy rate. The key benefit is the accuracy even with enormous datasets and relatively short training time. The model might not be appropriate for some regression tasks and can be computationally intensive for large datasets and can require significant training time.

5.1.5 CATBOOST

This updated ensembler can handle categorical features utilising sorted target statistics rather than one-hot encoding. The greedy technique takes the aim for a category group and averages it. One advantageous characteristic of this method is its successful use with default parameters and decision tree, which reduces the time required for prediction and parameter adjustment. Nevertheless, target leakage occurs because the target value is utilised to construct a representation for the categorical variables, which is then used for prediction. Based on the data, CB can automatically choose the ideal number of trees and manage missing values. Although there are various built-in methods for tuning the hyperparameters, it can be sensitive to the selection of the parameters. Its applications include ranking, recommendation systems, forecasting, and even personal assistants.

4.1.6 LIGHTGBM

To address the constraints of histogram-based techniques, which are typically utilised in all relevant frameworks, Light Gradient Boosting Machine utilises Gradient-based One Side Sampling and Exclusive Feature Bundling (EFB). The fundamental distinction in decision trees' construction is that the tree is divided leaf-wise, with the leaf with the largest delta loss being chosen to grow. To handle categorical characteristics, LightGBM employs a unique technique that combines one-hot encoding with the gradient-based approach. The capacity to manage missing values, reduce memory use and training time by of the histogram-based feature binning and automatically choose the ideal number of trees depending on the data are just a few of this gradient boosting method.

4.1.7 XGBOOST

In eXtreme Gradient Boosting, weights are allocated to all individualistic variables and then supplied into a decision tree that forecasts results. Those predicted inaccurately by the tree are raised, and these factors are then sent into the second decision tree. Individual classifiers/predictors are then used. These individual classifiers/predictors are then amalgamated to create a more powerful and precise model. Regularization and handling of missing data are done by generally characterised them as hyperparameters in the objective function. Weighted quantile sketch is a new addition that speeds up the algorithm's training process and uses less memory. Due to its robust and accurate nature, its application vary including problems involving regression, classification, ranking, and user-defined prediction.

5.2 Proposed Model

Our application uses deep feed forward, a type of artificial neural network. The algorithm is based on the principle that inputs labelled with weights are passed through multiple hidden layers before being computed together and compared to the threshold value. Back-propagation is a process in which the weights of the dataset are altered using the delta rule based on the output. The general architecture layers are made up of neurons, activation functions, sigmoid, tanh, and rectified linear units, as well as input, output, and hidden layers.

The neural network operates in a single direction with no cycles, with each layer of neurons performing a nonlinear transformation on the input and feeding the output to the next layer. Commencing with the input layer, it functions as a neural bridge to the hidden layer composed of neurons that apply a weighted sum of the inputs to a nonlinear activation function. It results in the neuron's output, which is then adjusted through back propagation. The activation function employed depends on the issue; for example, the sigmoid function is widely used in binary classification issues, whereas the softmax function is utilised in multiclass classification problems. For regression issues, the output layer is typically built around one neuron with a linear function. Image categorization, natural language processing, and speech recognition are just a few of the applications for the DFF model. Nevertheless, if the network is too vast or the training data is noisy, they may suffer from overfitting which can be prevented through regularization techniques like as dropout or weight decay.

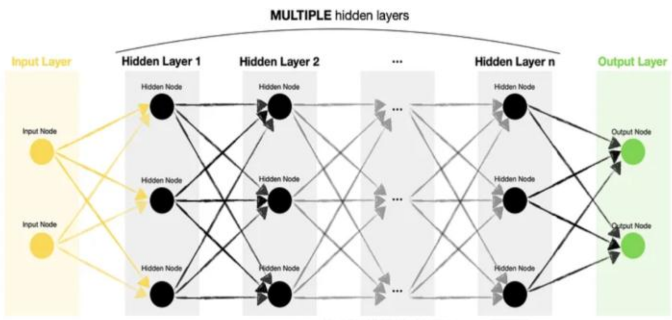


Fig -10: DFF MODEL ARCHITECTURE

The algorithm dataset was chosen from kaggle as a training set with 8,878 records and 328 features and a testing set with 4,760 records and 328 features for our application. Both of which are pre-processed to exclude inappropriate and null data and adjust features and records to ensure the performance efficiency of the algorithm. After conversion and universalization of content, removal of redundant constant features, the desired features are selected based on univariate selection and feature importance. The former uses SelectKBest class, provided by scikit-learn library, to find the top features with strongest link to the output and the latter uses Extra Tree classifier to score its relevance toward the output. After which, the dataset is split into training and testing batches where train size comprises of 80% data. Using the scaler object created by StandardScaler, the data are scaled and reshaped according to the input type. DFF (deep feed forward) model is initiated with input layer of 25 selected features. Following which are 5 hidden layers each with a specified dense layer of varied neurons frequency, activation of rectified linear units and HeNormal kernel initializer. To counteract over-fitting the training dataset, a dropout rate of 0.4 is employed to discard 40% of the nodes' material. The output dense layer uses softmax activation consists of 5 neurons, because of the presence of 4 levels of urgency.

```
Model: "DFF-Model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 Hidden-Layer-1 (Dense)      (None, 256)               7680

 dropout (Dropout)           (None, 256)               0

 batch_normalization (BatchN  (None, 256)              1024
 ormalization)

 Hidden-Layer-2 (Dense)      (None, 128)               32896

 dropout_1 (Dropout)         (None, 128)               0

 batch_normalization_1 (Batc  (None, 128)              512
 hNormalization)

 Hidden-Layer-3 (Dense)      (None, 64)                8256

 dropout_2 (Dropout)         (None, 64)                0

 batch_normalization_2 (Batc  (None, 64)               256
 hNormalization)

 Hidden-Layer-4 (Dense)      (None, 32)                2080

 dropout_3 (Dropout)         (None, 32)                0

 batch_normalization_3 (Batc  (None, 32)               128
 hNormalization)

 Hidden-Layer-5 (Dense)      (None, 16)                528

 Output-Layer (Dense)        (None, 5)                 85

=================================================================
Total params: 53,445
Trainable params: 52,485
Non-trainable params: 960
```

Fig -11: DFF Model Description

These callbacks in Keras used for monitoring the training process and preventing overfitting, learning rate will be reduced if the validation loss does not improve for 20 epochs, and the training process will be stopped if the validation loss does not improve for 60 epochs. The weights of the best-performing model during training will be restored

```
from tensorflow.keras.callbacks import ReduceLROnPlateau, EarlyStopping
reduce_lr = ReduceLROnPlateau(
    monitor="val_loss",
    factor=0.8,
    patience=20,
)
early_stop = EarlyStopping(
    monitor="val_loss",
    patience=60,
    restore_best_weights=True
)
callbacks = [reduce_lr, early_stop]
```

```
model.compile(optimizer = 'adam', loss = 'sparse_categorical_crossentropy', metrics = ['accuracy'])
```

```
history=model.fit(reshaped_X_train_scaled, Y_train, batch_size = 128,validation_split=0.2, epochs = 300,verbose = 1,
              callbacks=[reduce_lr, early_stop])
```

Fig -12: Callback Function

6. Results (comparison)

Using the Deep Feed Forward model, the dataset of was trained of 13,638 records and 329 features to ultimately rank each instance from 1 to 4. The sequential model using the output dense layer through softmax activation gives the output accuracy. The model has a total of 53,445 params in which 52,485 are trainable and the rest are non-trainable. The DFF model uses the adam optimizer to compile its accuracy. Through 300 epoch with batch size of 128, the model was evaluated to have a loss score of 34.26% and accuracy of 88.40%.

```
score, acc = model.evaluate(reshaped_X_test_scaled, Y_test, verbose=1)
print('loss score:', score*100)
print('accuracy:', acc*100)
```

```
56/56 [==============================] - 0s 4ms/step - loss: 0.3426 - accuracy: 0.8840
loss score: 34.25649106502533
accuracy: 88.40090036392212
```

Fig -13: CV score of DFF

In comparison with the other machine learning techniques, it is observed that DFF has the highest accuracy rate.

Model	Accuracy Rate
Deep Feed Forward (DFF)	**88.40 %**
XGBOOST	81.99%
LIGHTGBM(LGM)	81.48%
CATBOOST	81.01%
RANDOM FOREST (RF)	79.55%
DECISION TREE (DT)	77.06%
SUPPORT VECTOR MACHINE(SVM)	70.44%
K-Nearest Neighbors (KNN)	69.80%

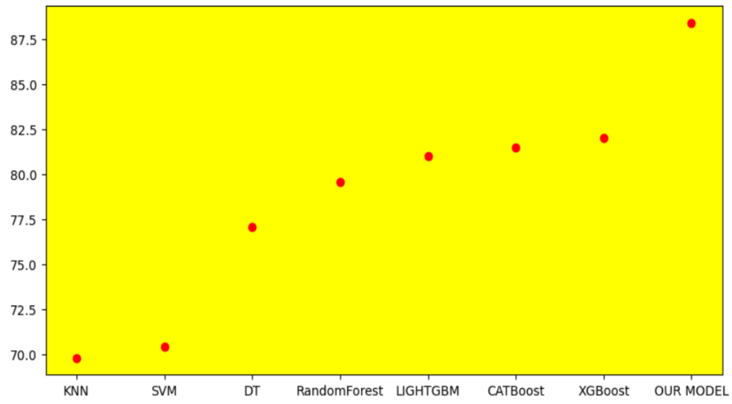


Fig -14: Algorithm Comparison Result

SYSTEM OVERVIEW

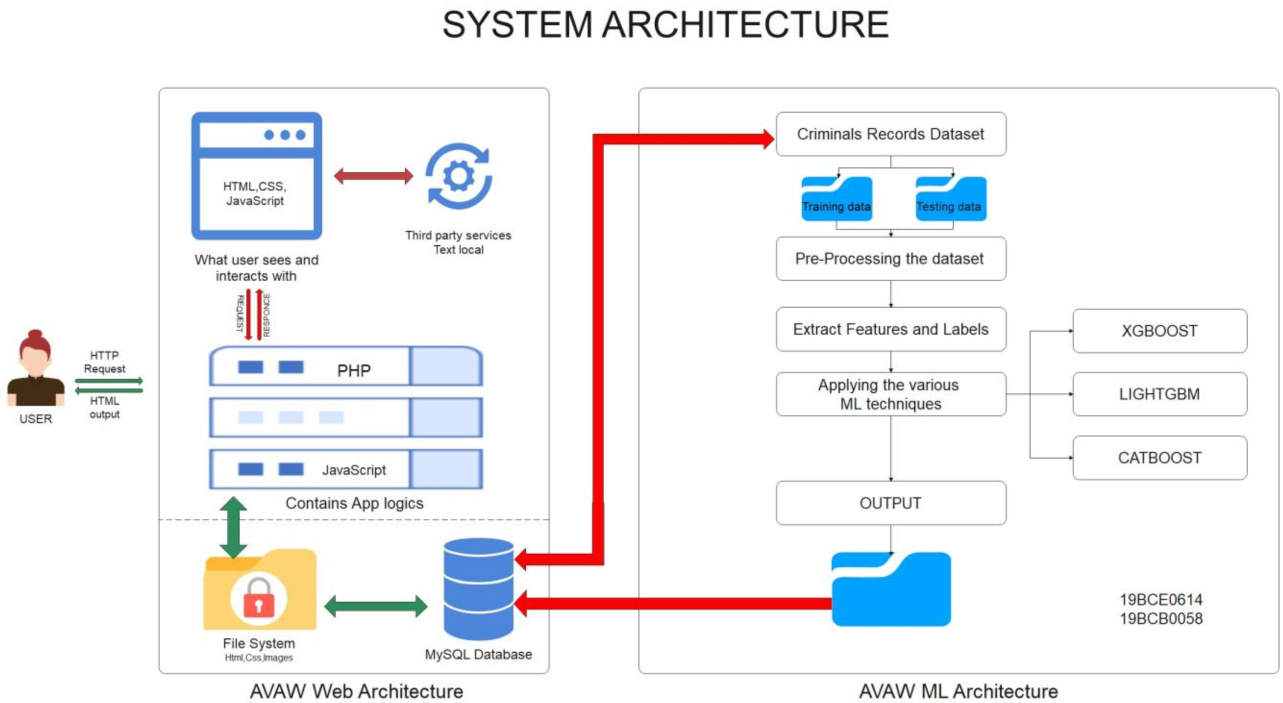


Fig -15: System Architecture

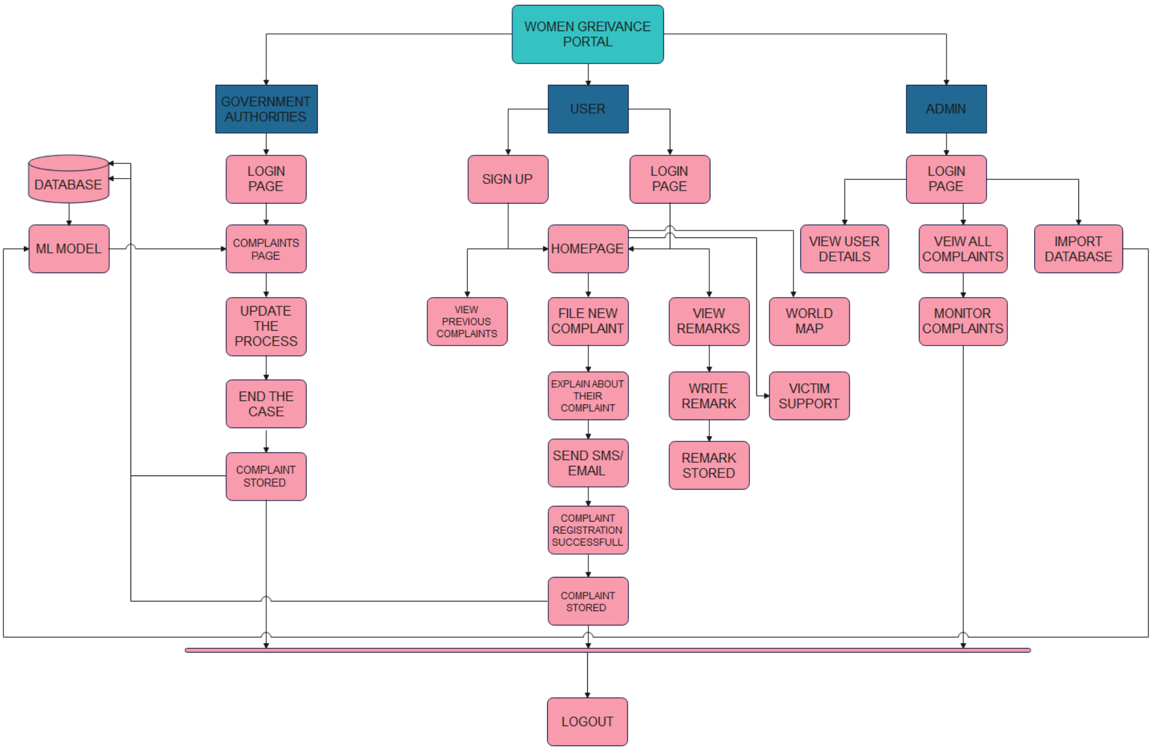


Fig -16: Block Diagram of Portal

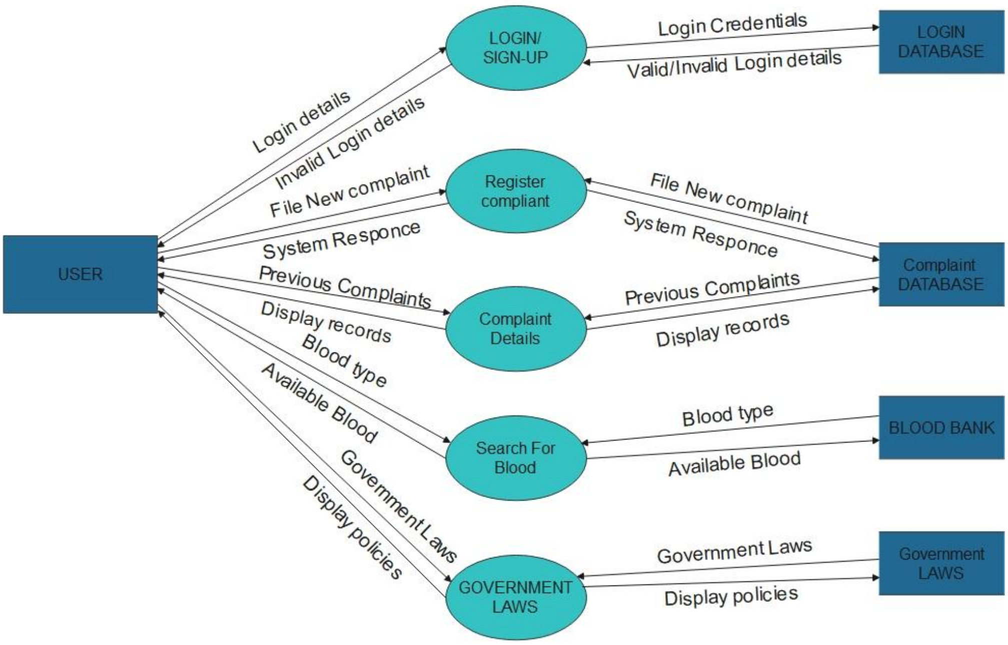


Fig -17: Dataflow Diagram

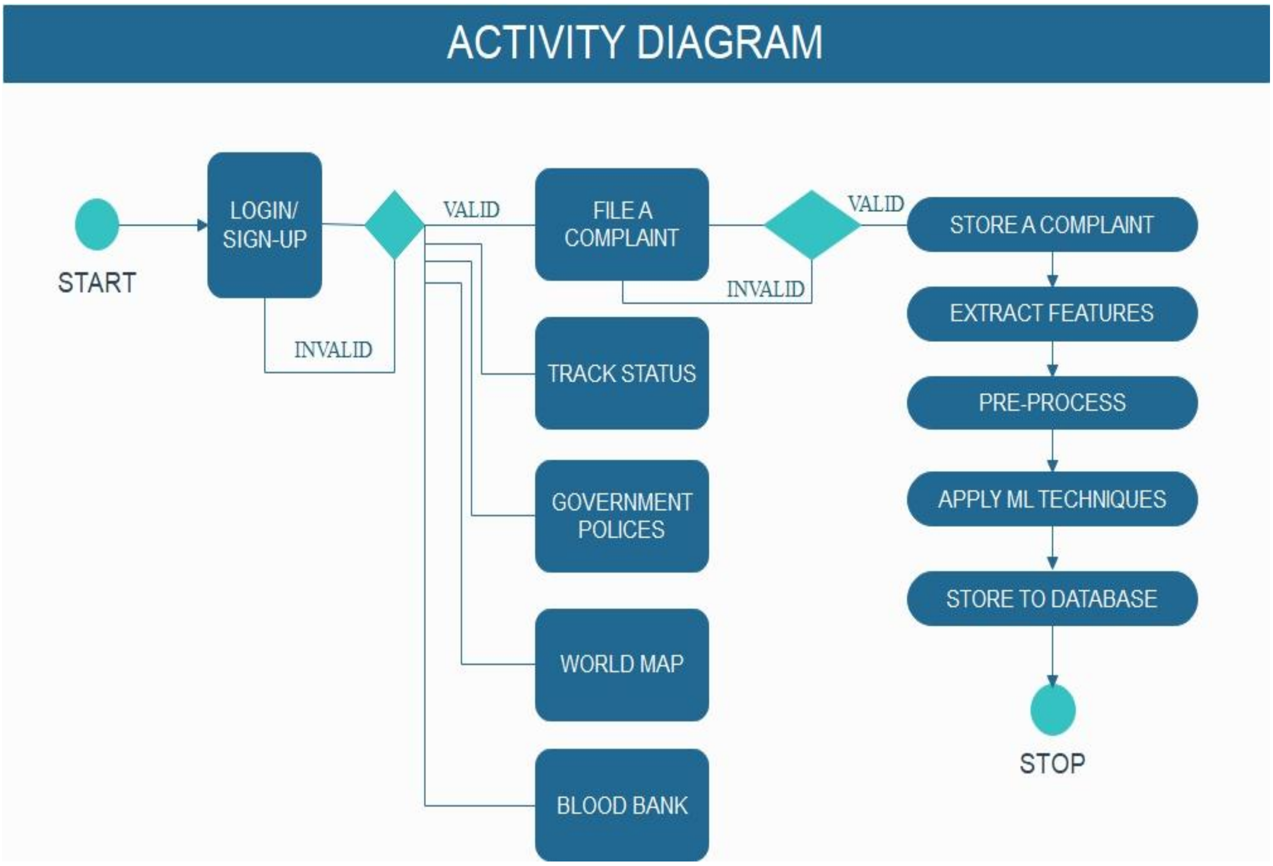


Fig -18: Activity Diagram

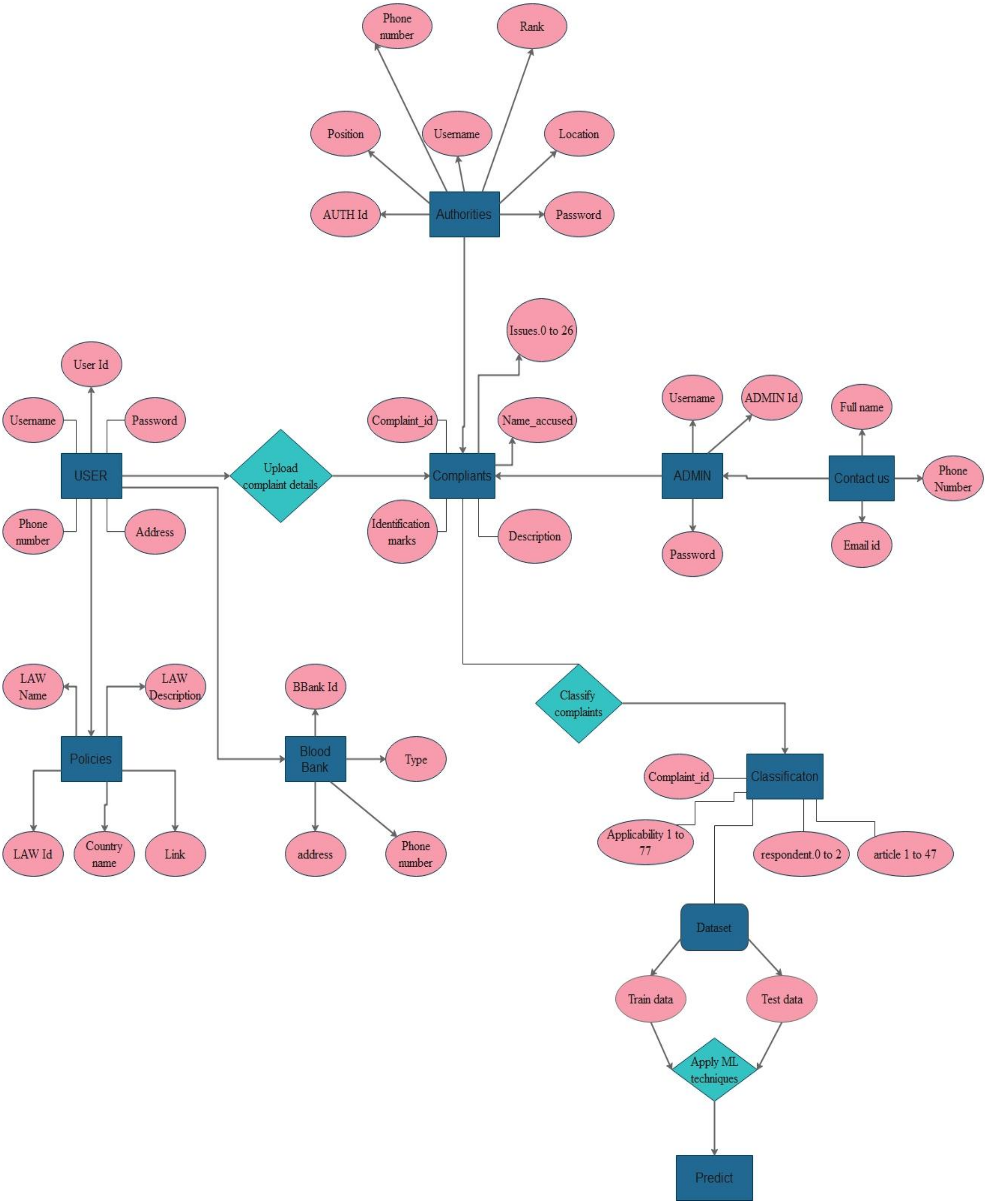


Fig -19: Entity Relationship Diagram

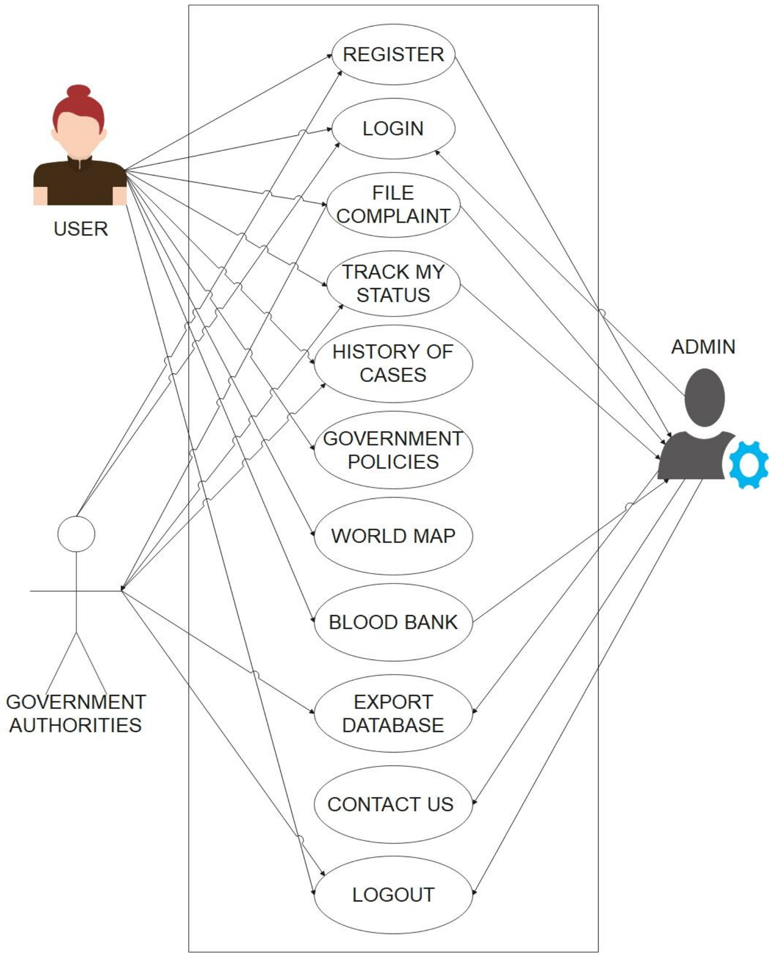


Fig -20: Usecase Diagram

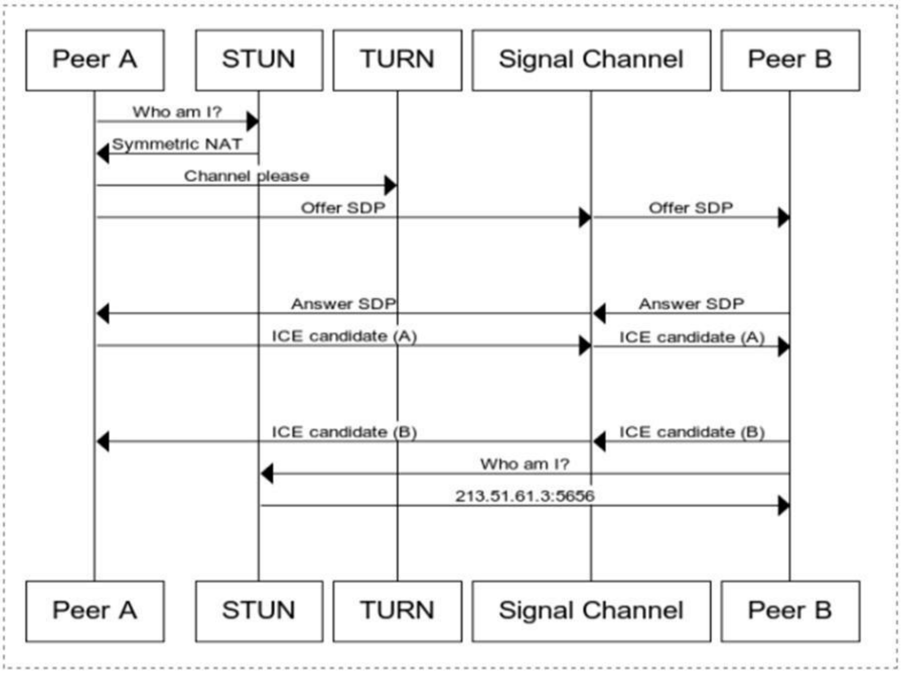


Fig -21: WebRTC Sequence Diagram

Web application Front-end:

INDEX HOMEPAGE

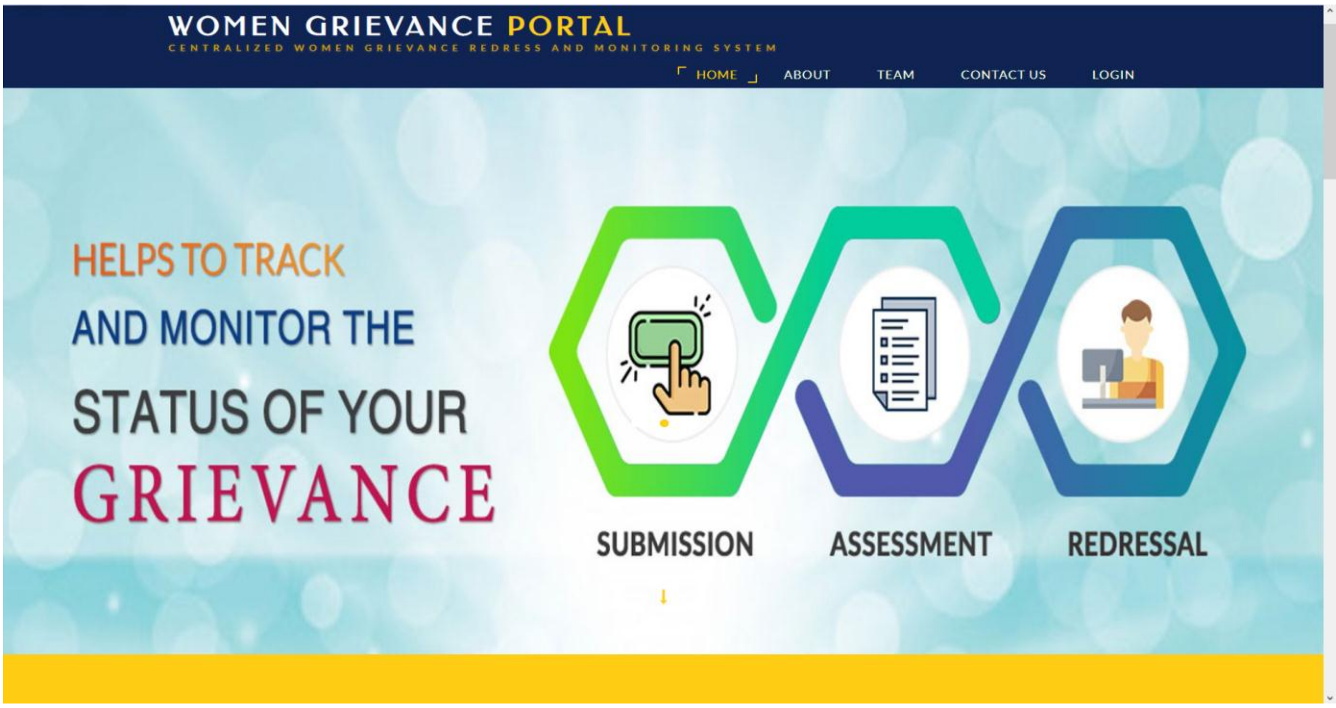


Fig -22: Homepage of Portal-Banner

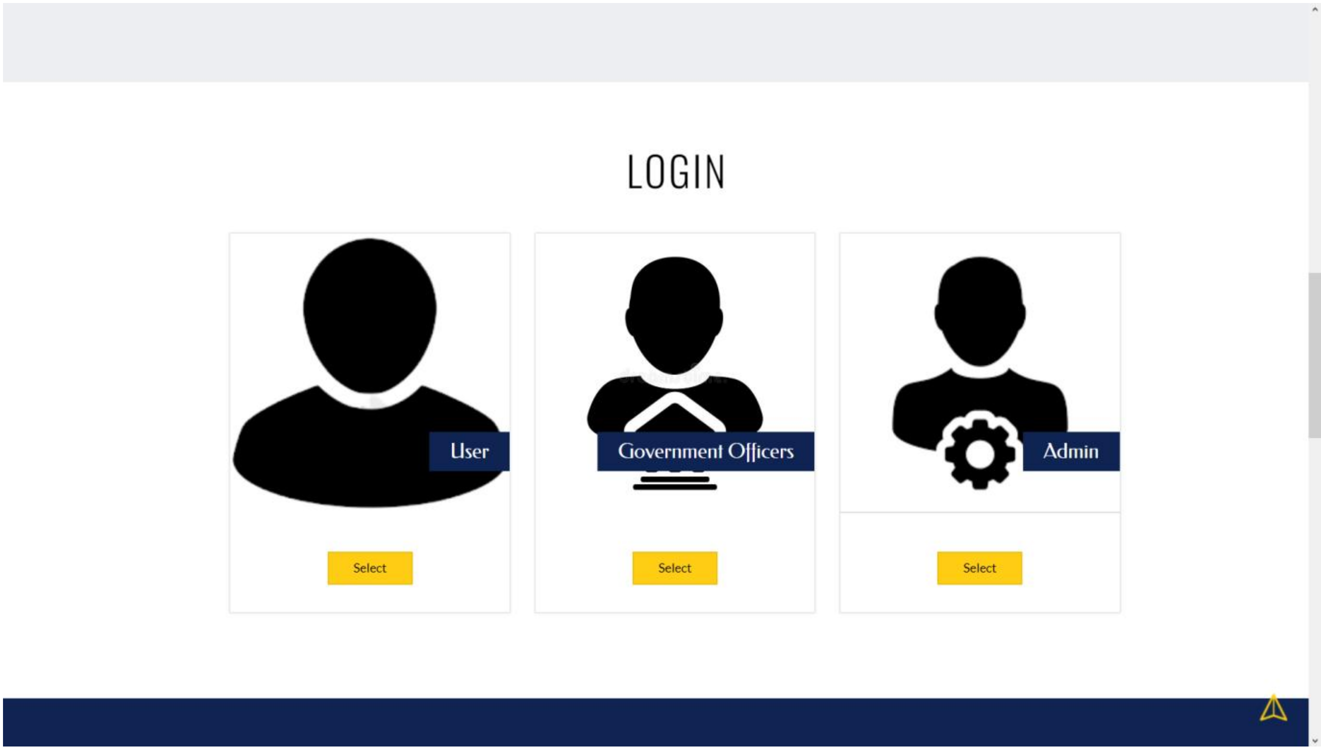


Fig -23: Homepage of Portal- Login for Users

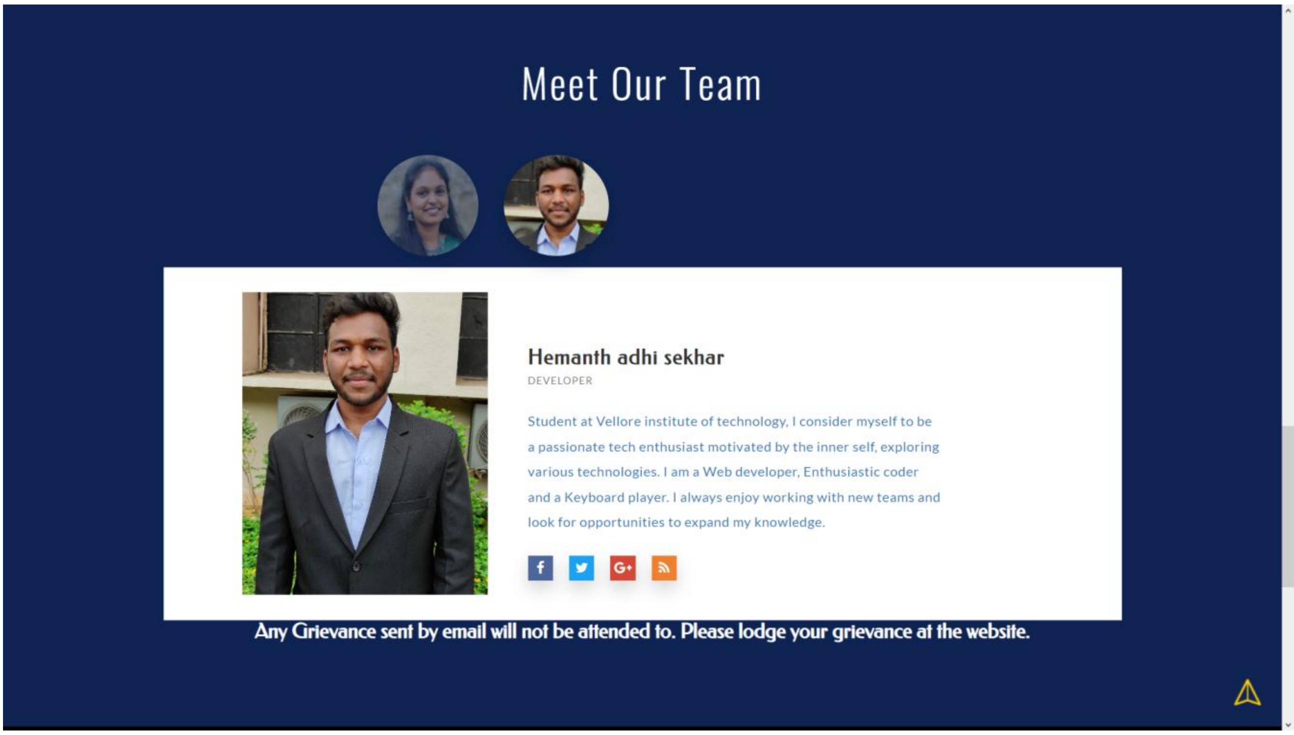


Fig -24: Homepage of Portal- Developer Team

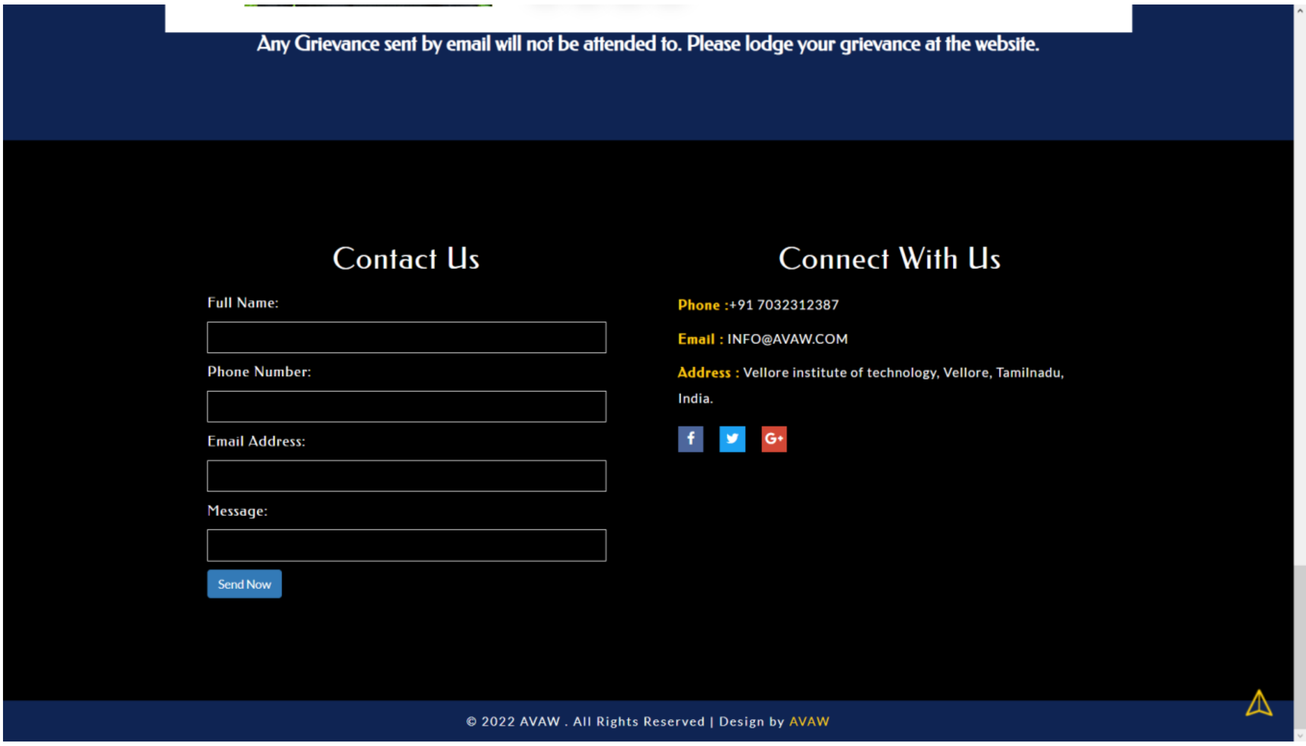


Fig -25: Homepage of Portal- Communication Details and Form

USER MODULE:

SIGN IN & LOGIN PAGE

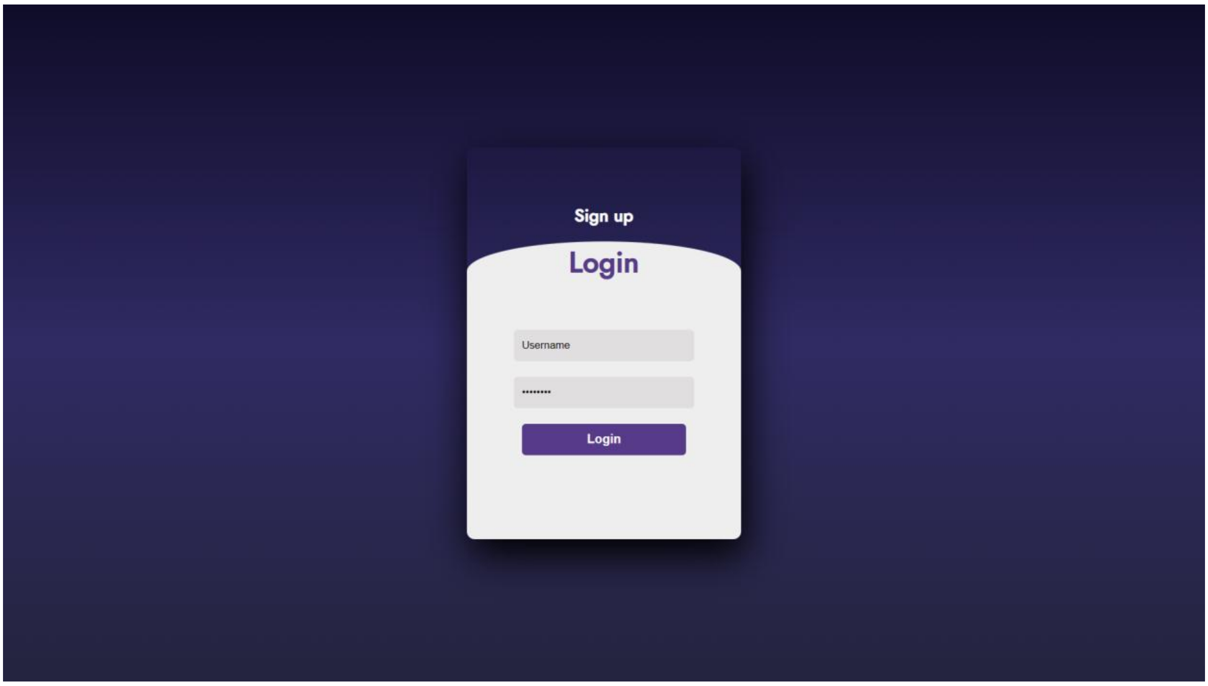


Fig -26: User Registration and Login Page

REGISTER PAGE

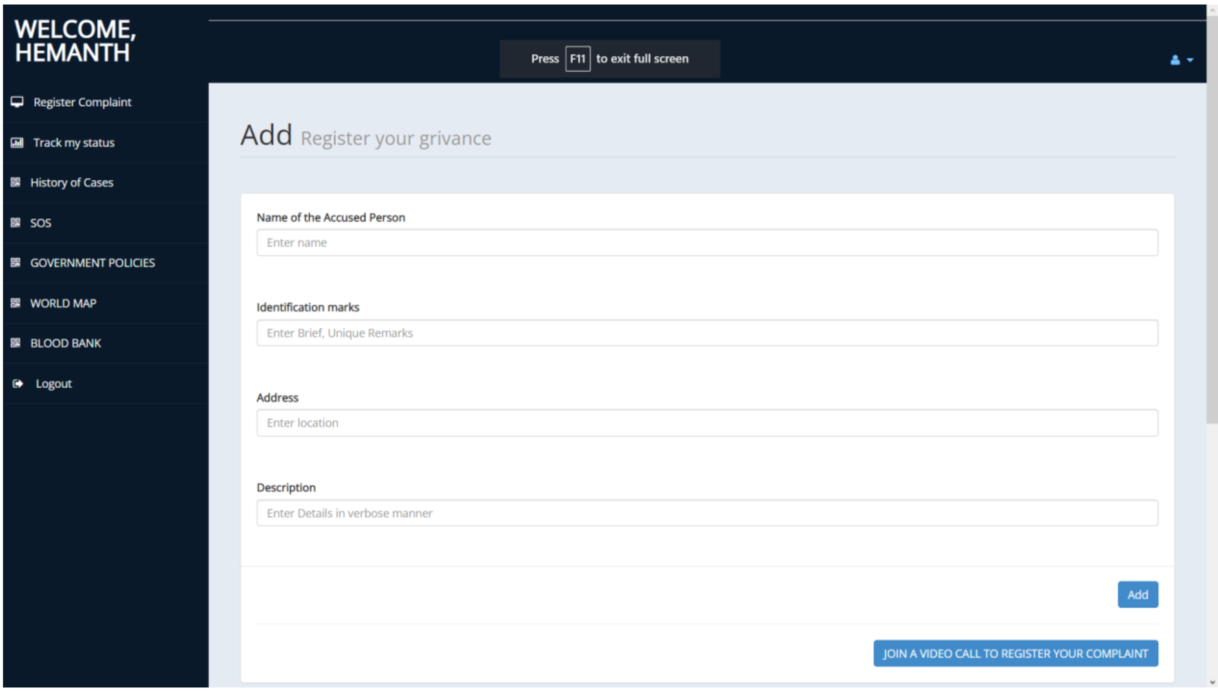


Fig -27: User Complaint Registration Page

TRACK STATUS PAGE

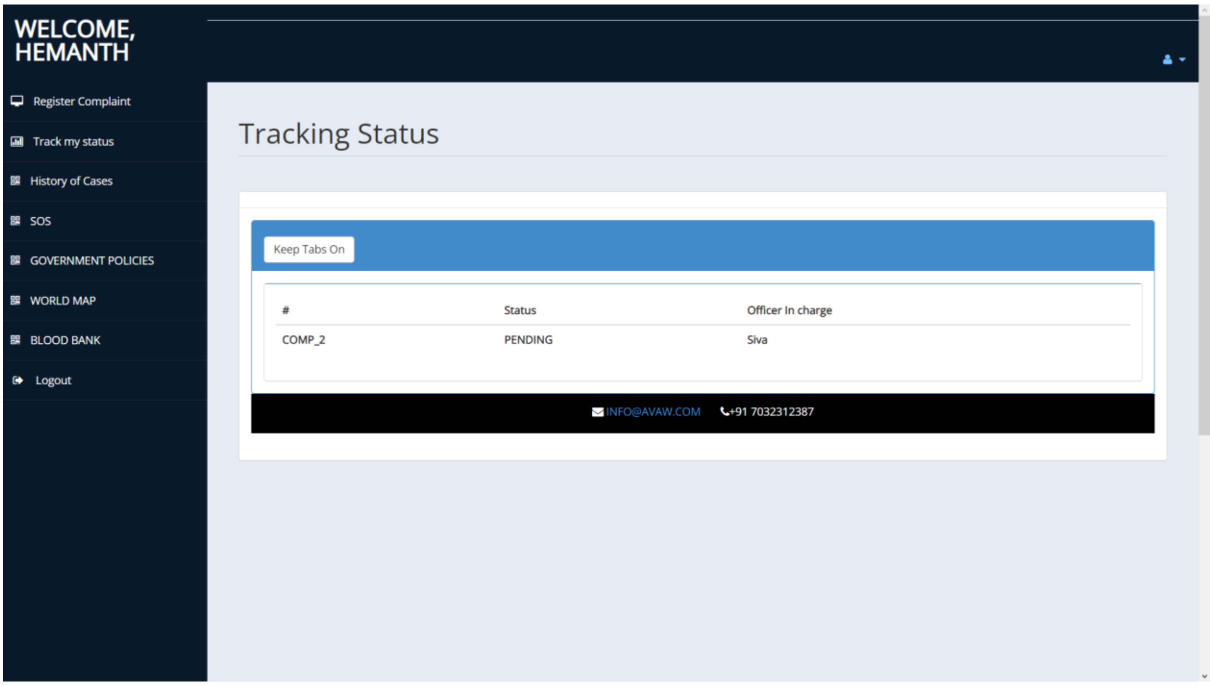


Fig -28: User Status Tracking Page

WORLD MAP PAGE

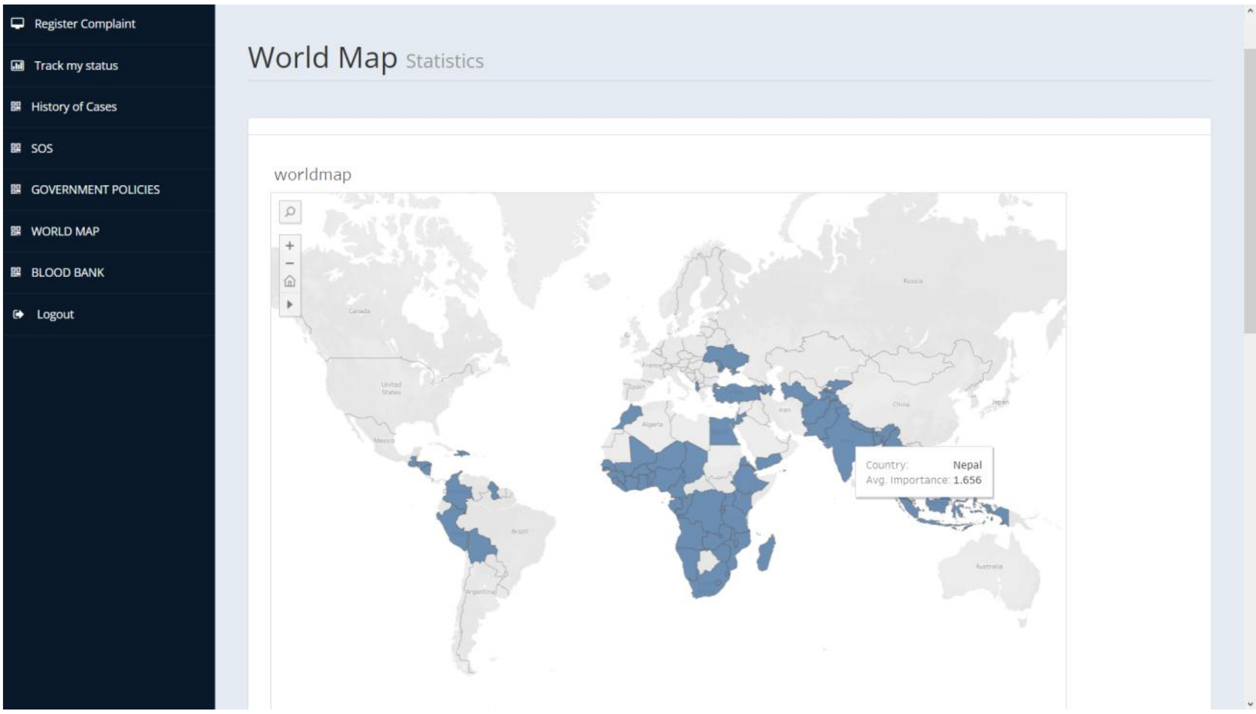


Fig -29: Global Violence Statistics Page

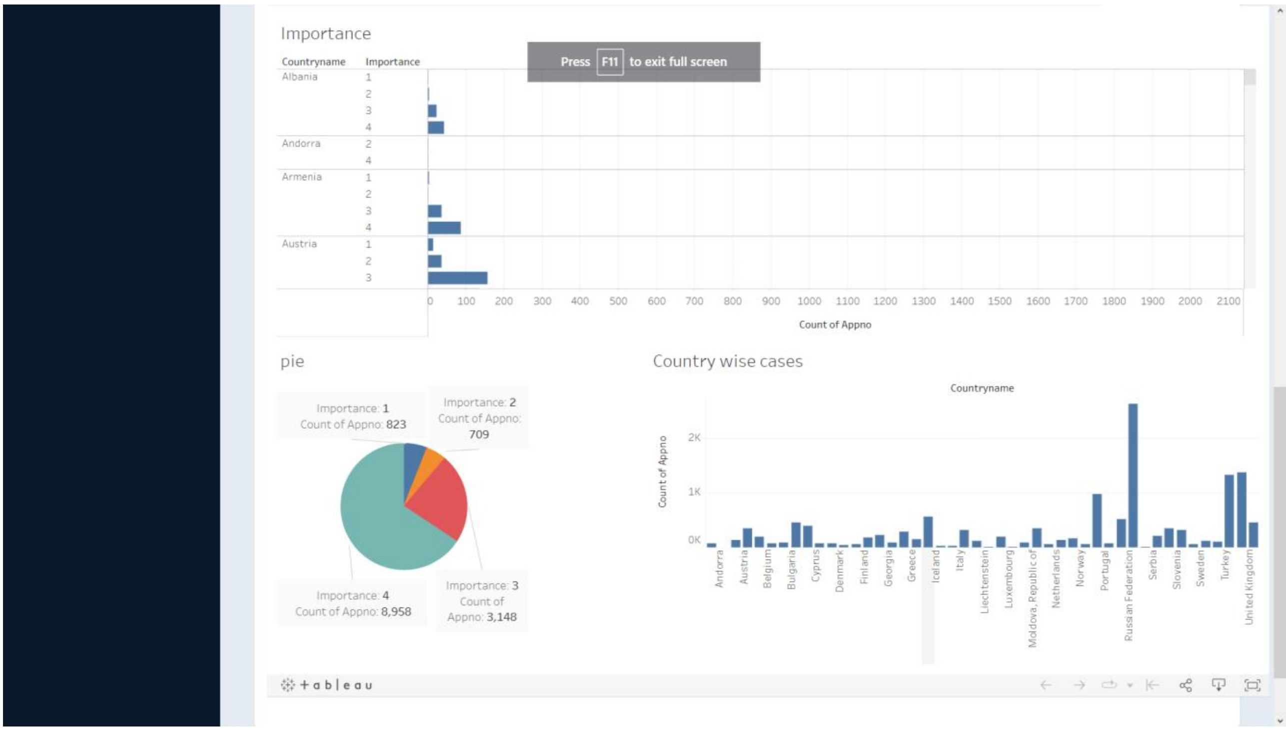


Fig -30: Global Violence Statistics Page

HISTORY OF CASES PAGE

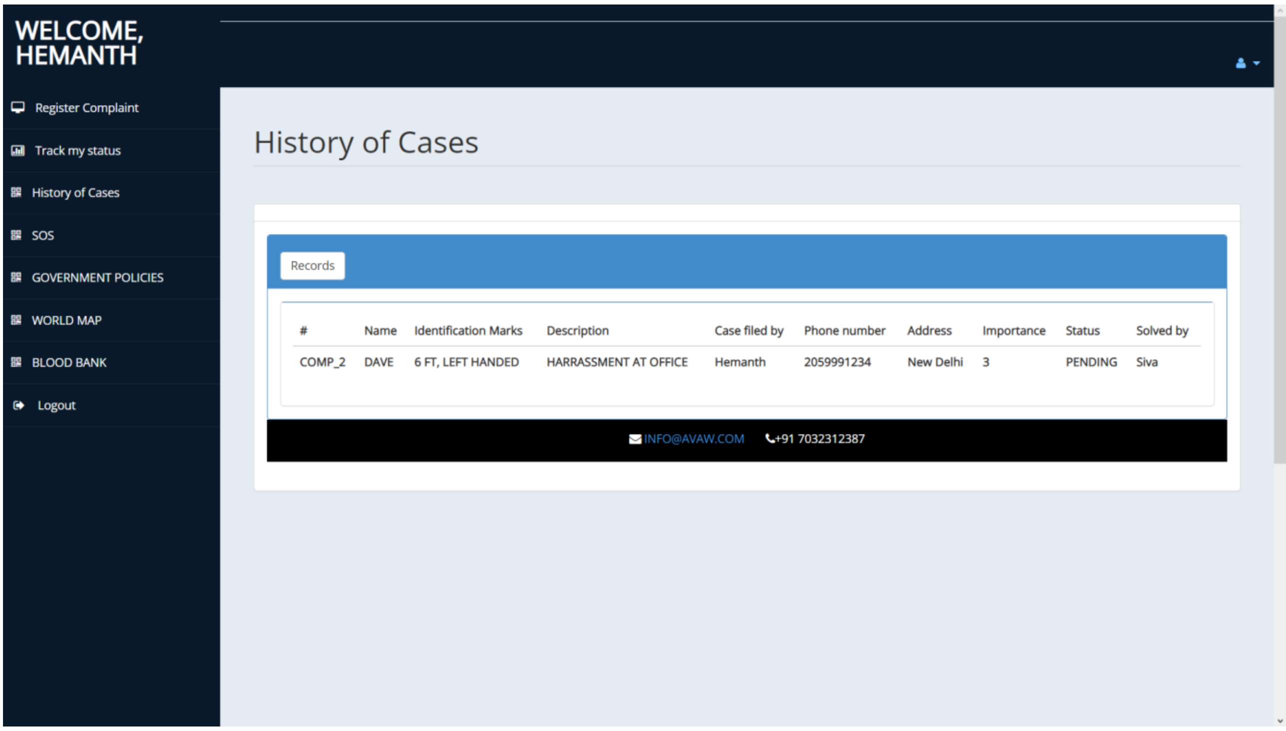


Fig -31: User History of Cases Submitted Page

LAW & POLICY PAGE

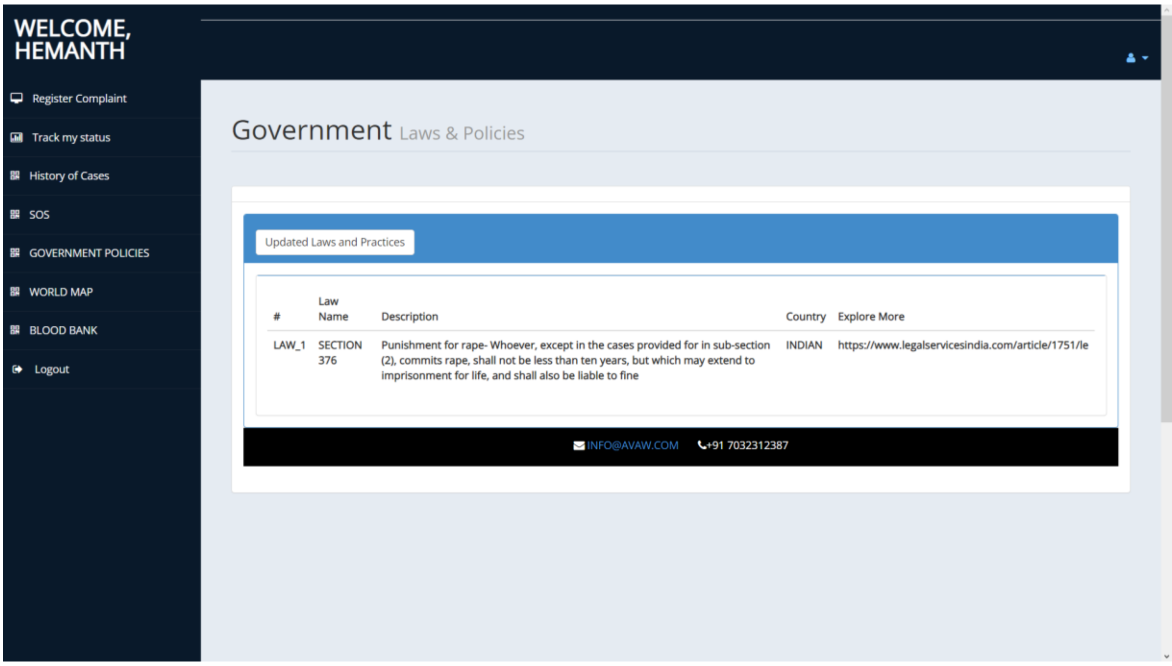


Fig -32: Government Laws and Policies Page

BLOOD BANK PAGE

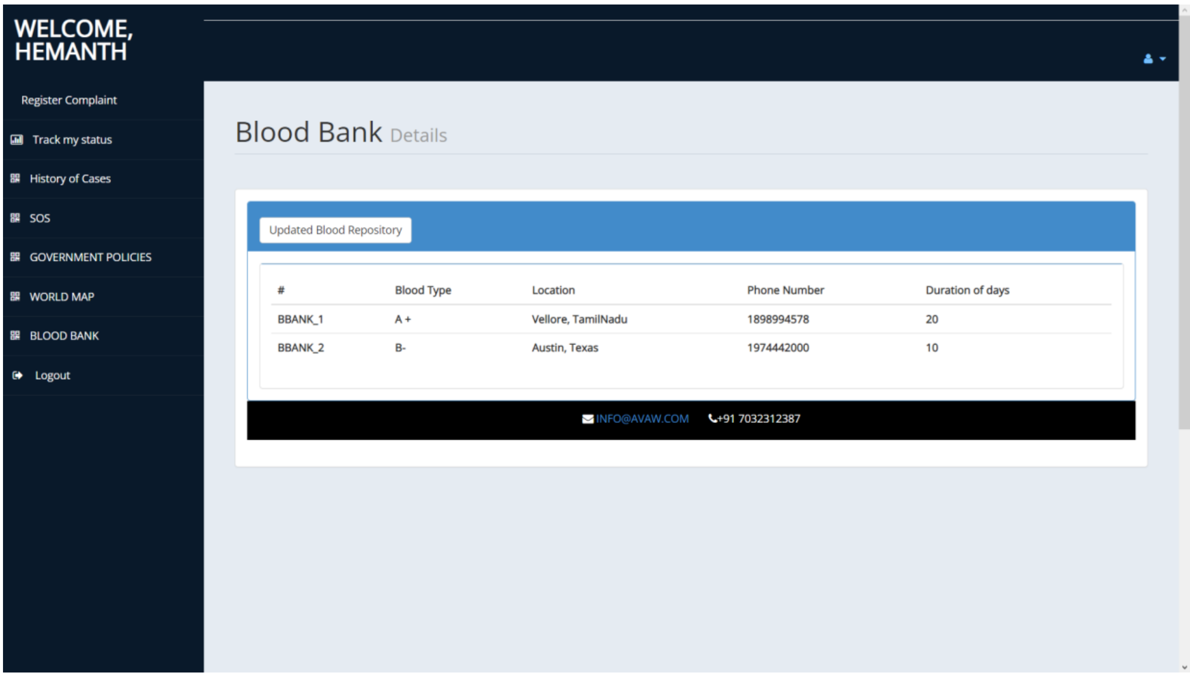


Fig -33: Blood Bank Repository Page

COMUNICATION PAGE

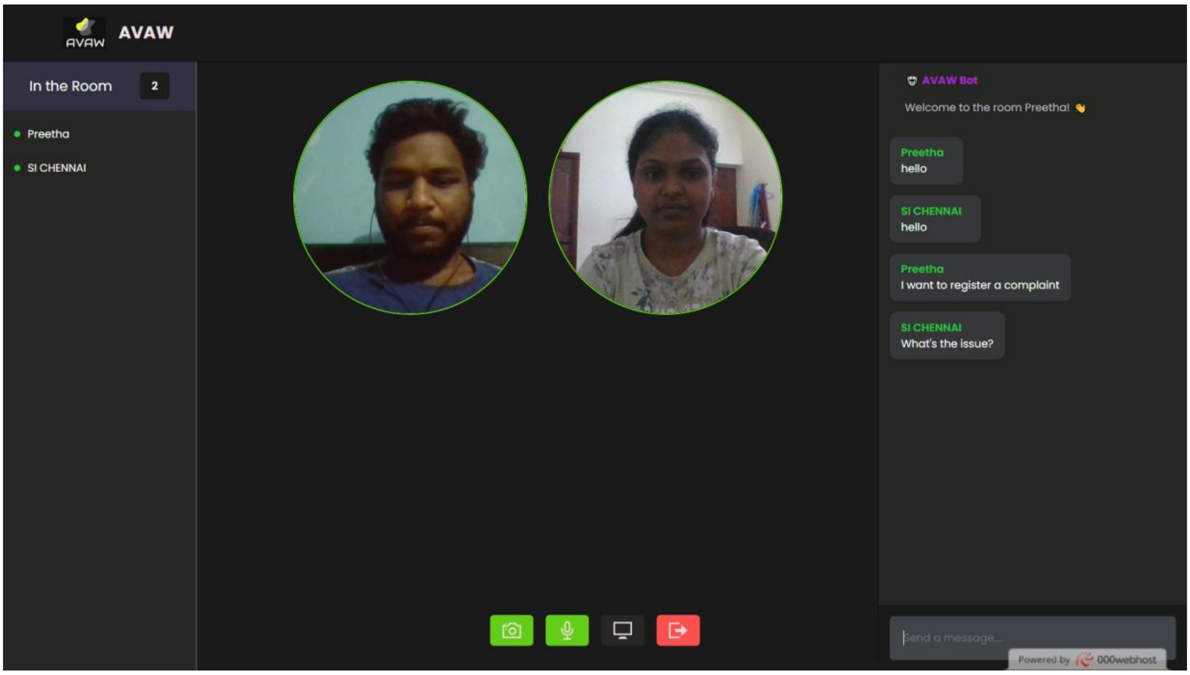


Fig -34: WebRTC Video Call Screenshot

PROFILE PAGE

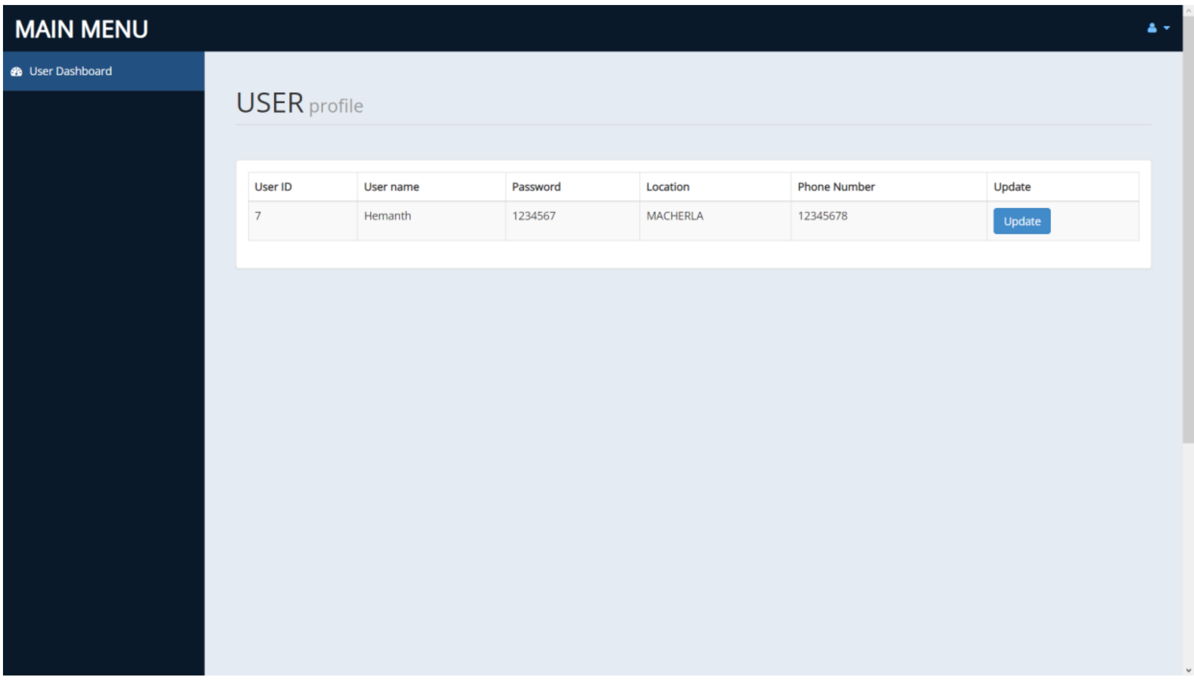


Fig -35: User Profile Page

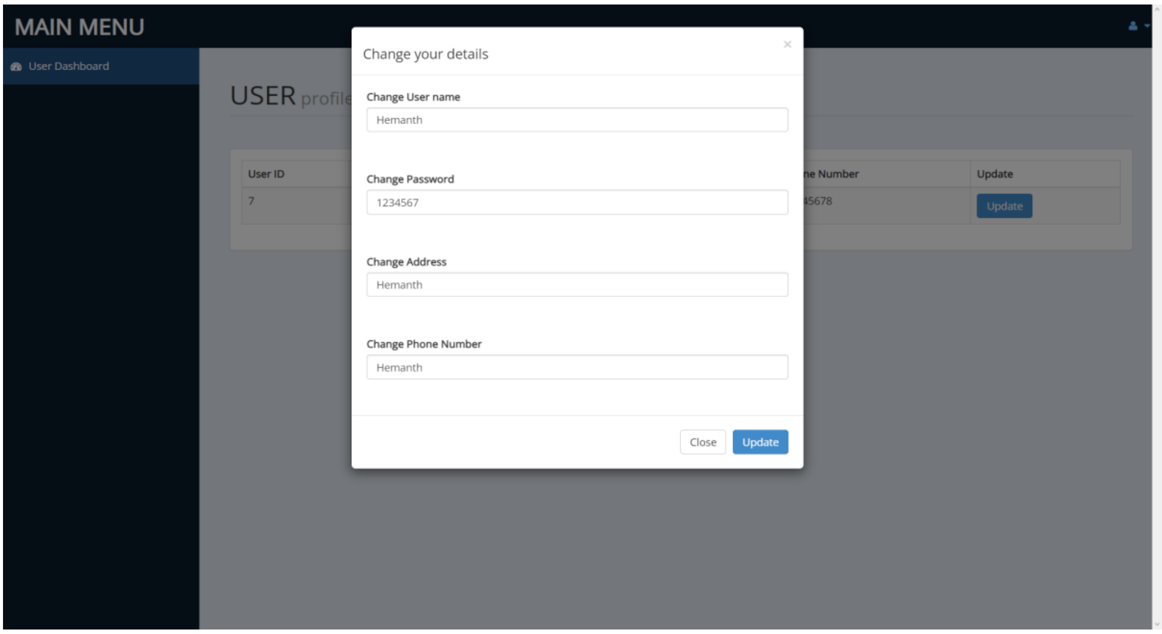


Fig -36: User Profile Updating Form

AUTHORITY MODULE:

Authorities are the government officials and they will have access to the functions accordingly after sign up and login process. They would be required to fill in legit information like their full name, phone number, location and position of authority. The system will redirect them to the landing page in which features like cases records, history of case handled by them, status tracking, database exporting and law and order database are available. Since users can video call to the nearby police station to communicate to a constable, police officers can join the room according and make a record of their case as an official complaint. Their profile can be edited.

CASE RECORD PAGE

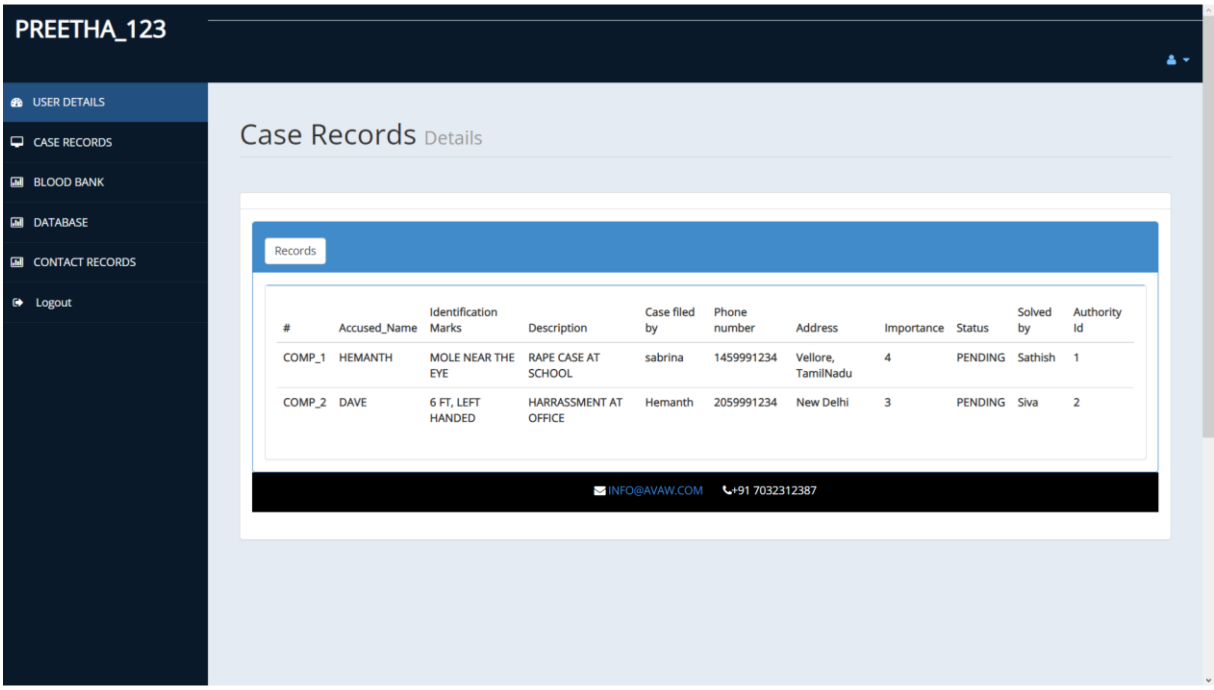


Fig -37: Case records Page

COMUNICATION PAGE

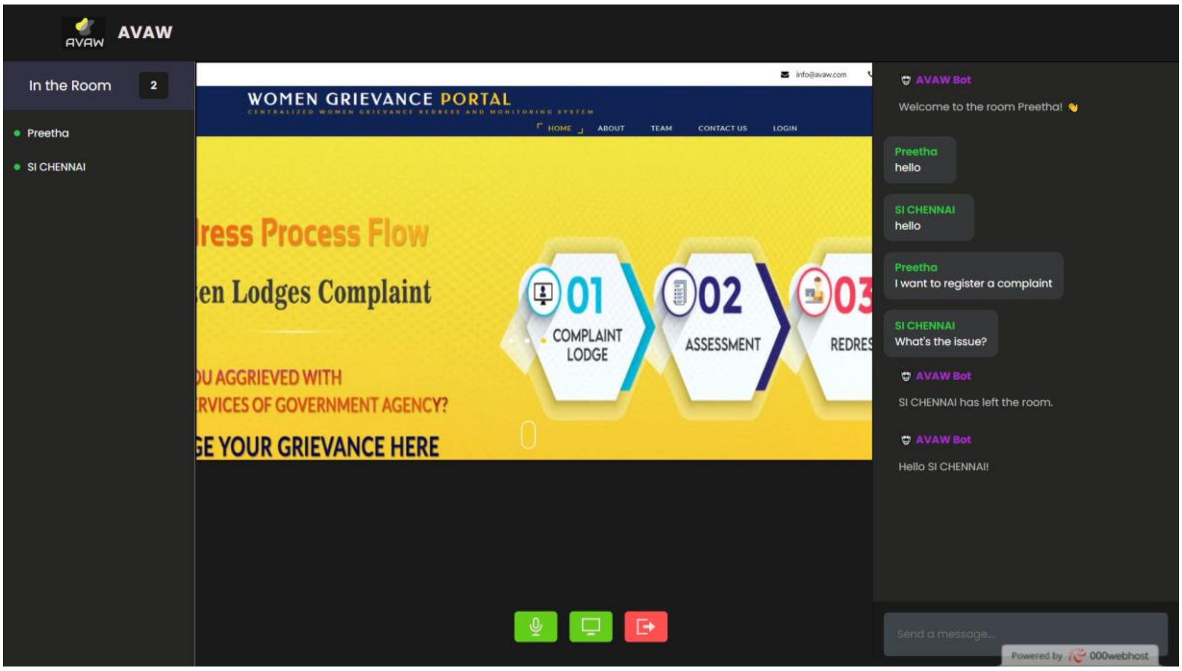


Fig -38: WebRTC Video Call Screenshot

EXPORT DATABASE PAGE

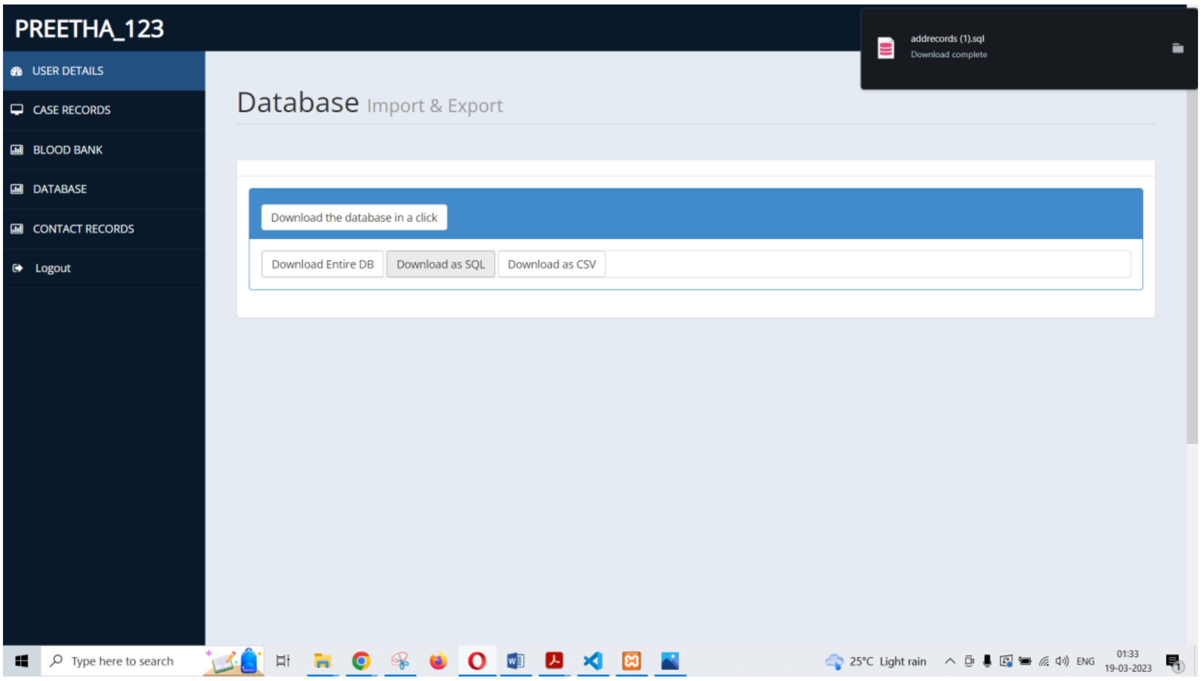


Fig -39: Database Exportation Page

ADMIN MODULE:

Admin is responsible for monitoring and managing all data storage and overall system. With proper authentication, admin has access to the landing page containing all features from viewing case records and user details to management of database and its exportation. The admin needs to filter out unnecessary or inappropriate information entered by either of the other users of the system.

USER DETAILS PAGE

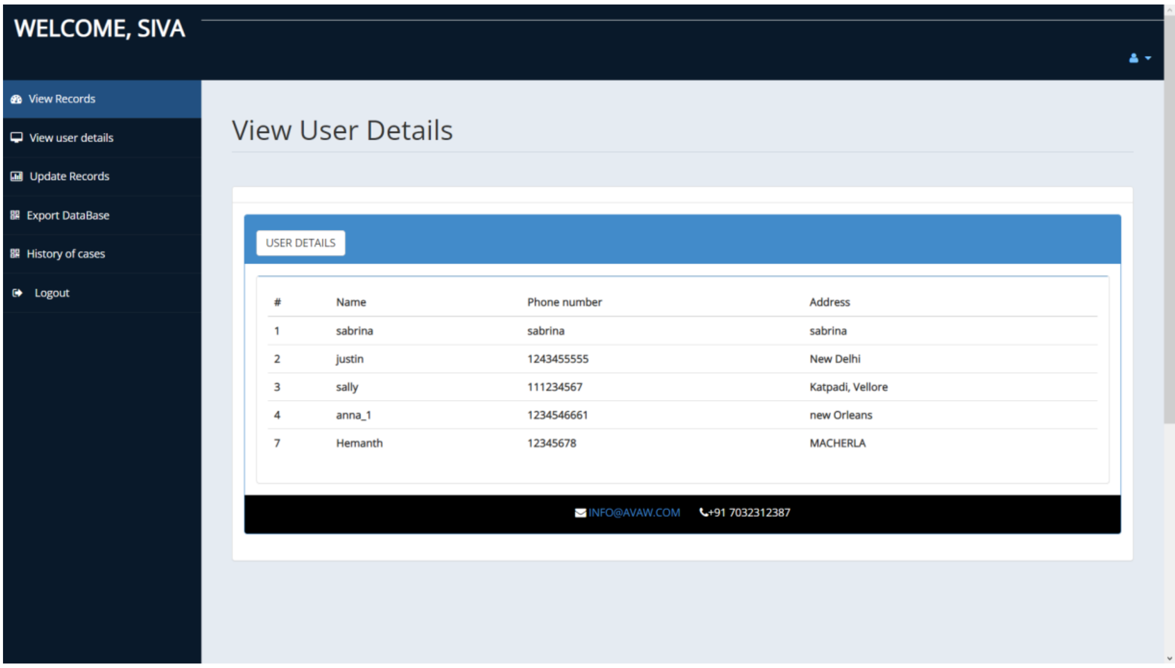


Fig -40: User Details Page

CONTACT FORM PAGE

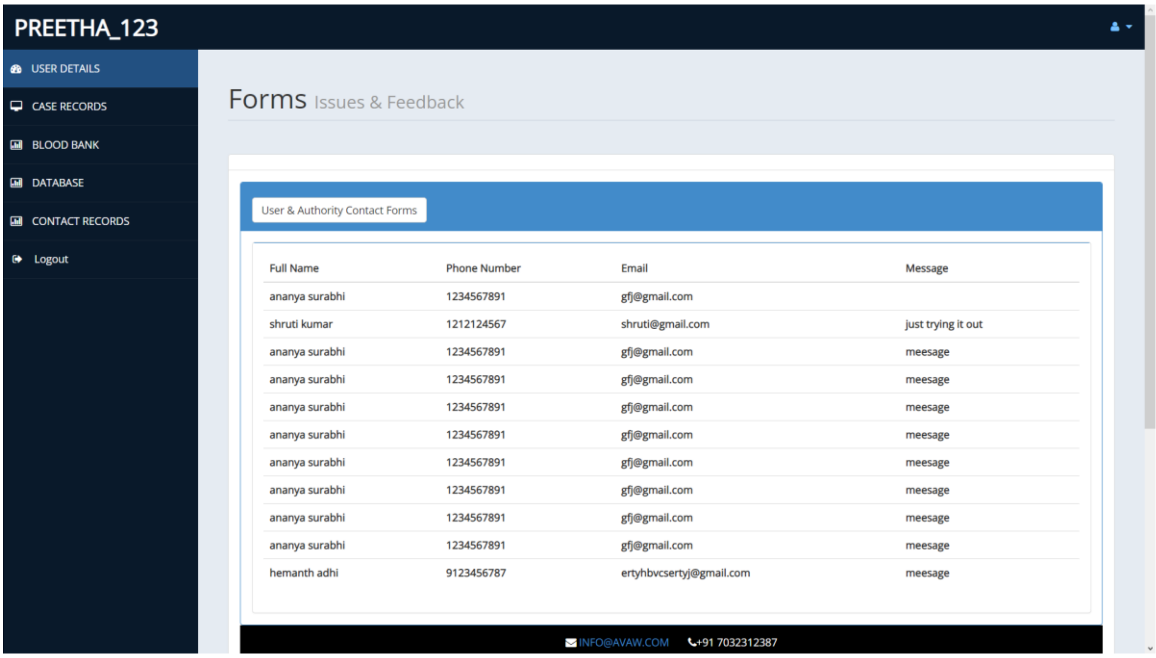


Fig -41: Contact Form Information Page

EXPORT DATABASE PAGE

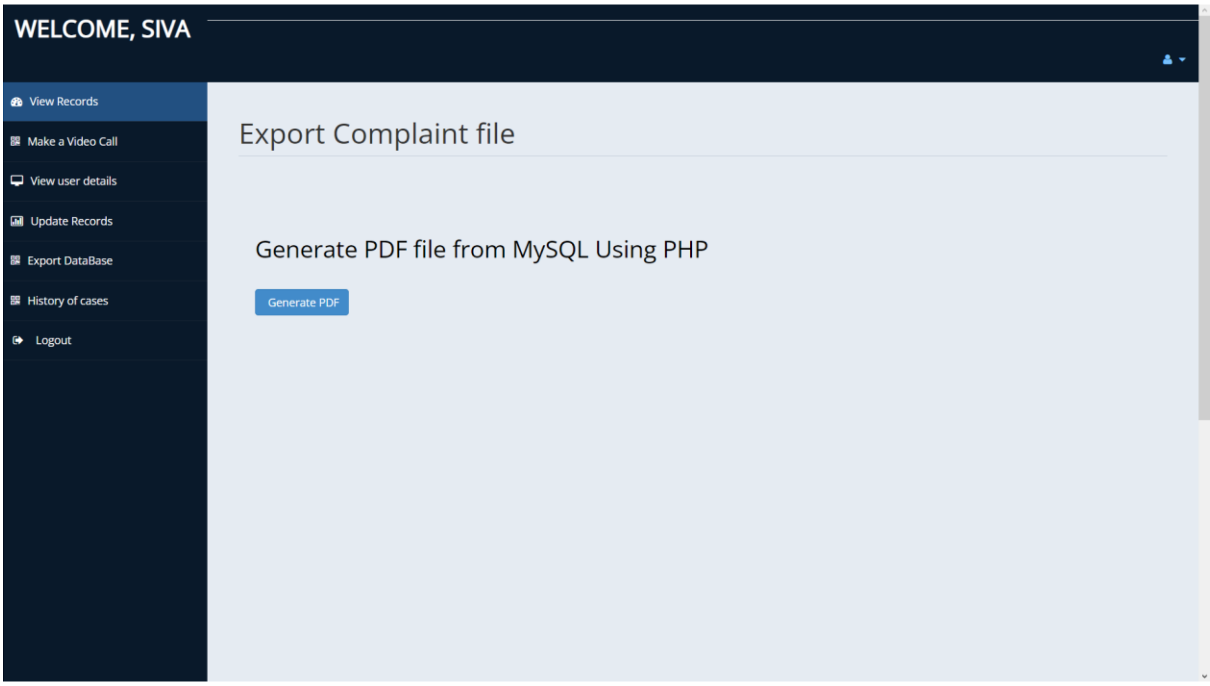


Fig -42: Database Exportation Page

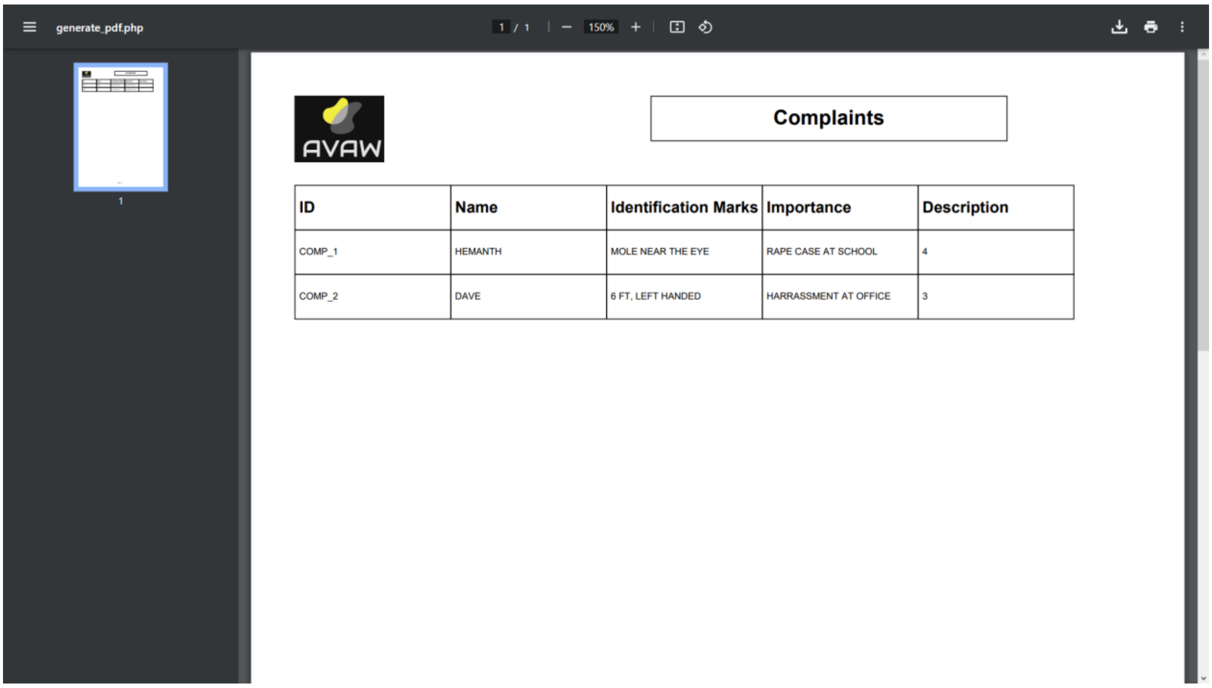
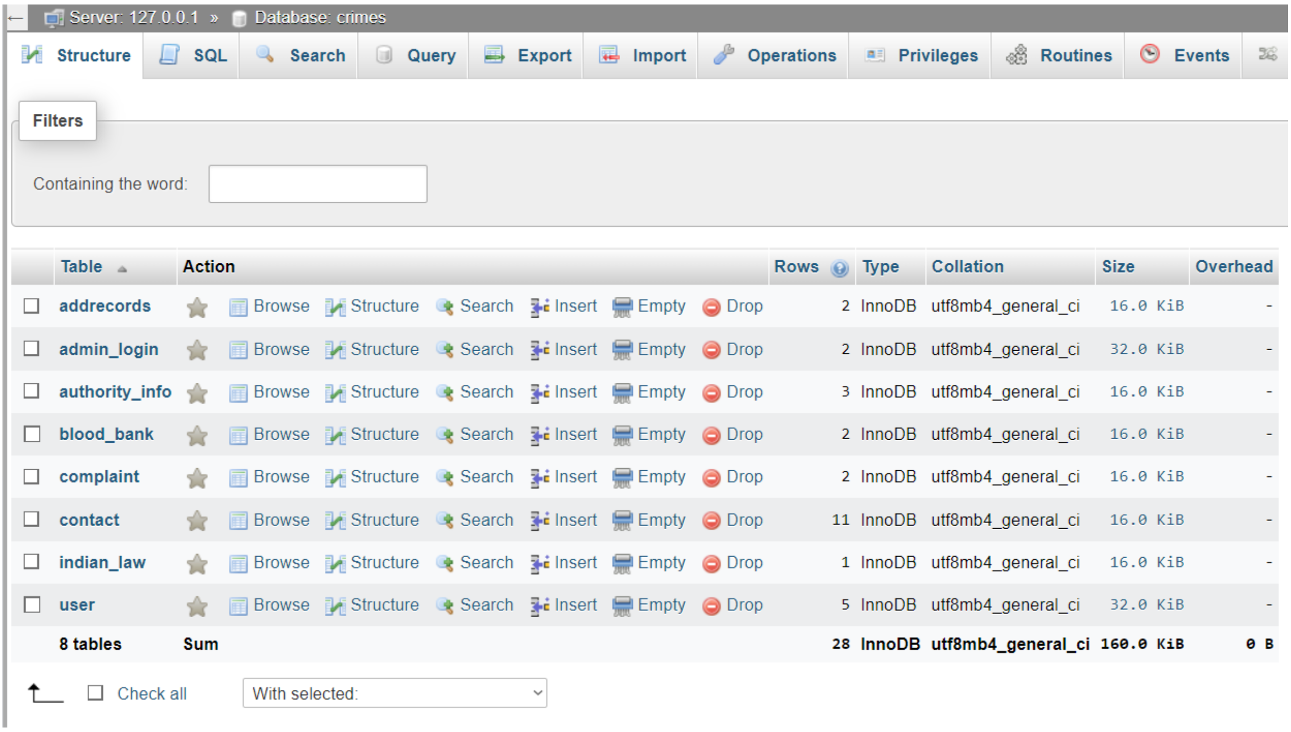


Fig -43: Generated PDF Screenshot

DATABASE


Server: 127.0.0.1 » Database: crimes

Structure | SQL | Search | Query | Export | Import | Operations | Privileges | Routines | Events

Filters

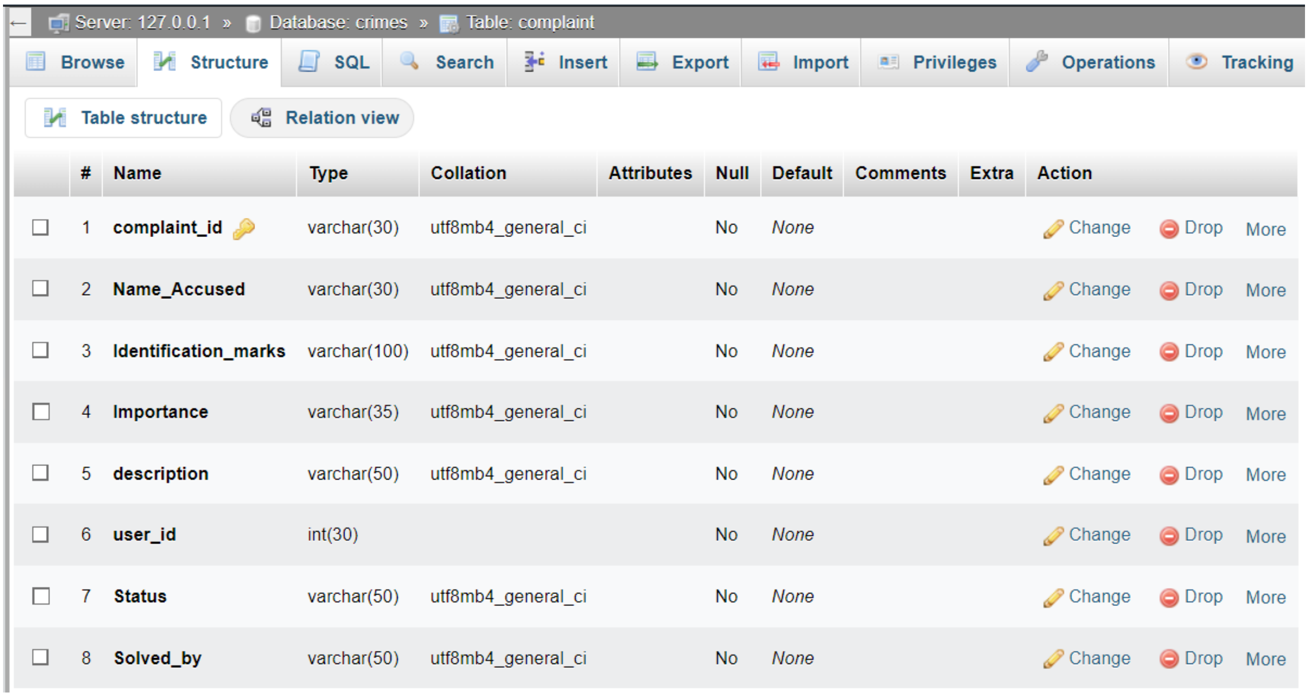
Containing the word:

Table	Action							Rows	Type	Collation	Size	Overhead
addrecords		Browse	Structure	Search	Insert	Empty	Drop	2	InnoDB	utf8mb4_general_ci	16.0 KiB	-
admin_login		Browse	Structure	Search	Insert	Empty	Drop	2	InnoDB	utf8mb4_general_ci	32.0 KiB	-
authority_info		Browse	Structure	Search	Insert	Empty	Drop	3	InnoDB	utf8mb4_general_ci	16.0 KiB	-
blood_bank		Browse	Structure	Search	Insert	Empty	Drop	2	InnoDB	utf8mb4_general_ci	16.0 KiB	-
complaint		Browse	Structure	Search	Insert	Empty	Drop	2	InnoDB	utf8mb4_general_ci	16.0 KiB	-
contact		Browse	Structure	Search	Insert	Empty	Drop	11	InnoDB	utf8mb4_general_ci	16.0 KiB	-
indian_law		Browse	Structure	Search	Insert	Empty	Drop	1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
user		Browse	Structure	Search	Insert	Empty	Drop	5	InnoDB	utf8mb4_general_ci	32.0 KiB	-
8 tables	Sum							28	InnoDB	utf8mb4_general_ci	160.0 KiB	0 B

Check all With selected:

Fig -44: List of Tables in Database

COMPLAINT TABLE-


Server: 127.0.0.1 » Database: crimes » Table: complaint

Browse | Structure | SQL | Search | Insert | Export | Import | Privileges | Operations | Tracking

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
1	complaint_id	varchar(30)	utf8mb4_general_ci		No	None			Change	Drop	More
2	Name_Accused	varchar(30)	utf8mb4_general_ci		No	None			Change	Drop	More
3	Identification_marks	varchar(100)	utf8mb4_general_ci		No	None			Change	Drop	More
4	Importance	varchar(35)	utf8mb4_general_ci		No	None			Change	Drop	More
5	description	varchar(50)	utf8mb4_general_ci		No	None			Change	Drop	More
6	user_id	int(30)			No	None			Change	Drop	More
7	Status	varchar(50)	utf8mb4_general_ci		No	None			Change	Drop	More
8	Solved_by	varchar(50)	utf8mb4_general_ci		No	None			Change	Drop	More

Fig -45: Attributes of Complaint Table in Database

CONCLUSION

Law enforcement departments are facing new challenges as crime rates continue to rise. They must keep their forces on the lookout for any signs of criminal activity, especially involving women. The goal of this application is to analyze and predict various types of crimes. The finished product would be a web application with the previously mentioned elements such as a user forum, a world map dashboard, services, and a government policy page. In the future, the application can be furthered developed by adding an illustration of the accused person in the complaint which can be sorted and analyzed by image recognition. In addition to the WebRTC, we can add options for voice translating services and speech to text conversion for users.

REFERENCES

[1] Mishra, Shivani & Kumar, Suraj. (2019). “A comparative study of crimes against women based on Machine Learning using Big Data techniques” Conference: International Conference on Distributed Computing and Internet Technology

[2] Vineet Pande, Viraj Samant, Sindhu Nair, 2016, Crime Detection using Data Mining, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 05, Issue 01 (January 2016), http://dx.doi.org/10.17577/IJERTV5IS010610

[3] P. Das and A. K. Das, "Crime analysis against women from online newspaper reports and an approach to apply it in dynamic environment," 2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDAC), Chirala, Andhra Pradesh, India, 2017, pp. 312-317, doi: 10.1109/ICBDACI.2017.8070855.

[4] Vinay Narayan Bhat, V Santhosh Kumar , Prof. Saravanan C ,”Analysis and Prediction of Crime against women in India using machine learning algorithms.", International Journal of Emerging Technologies and Innovative Research (www.jetir.org), ISSN:2349-5162, Vol.8, Issue 6, page no.a752-a758, June-2021, Available :http://www.jetir.org/papers/JETIR2106104.pdf

[5] M. R. Reza, F. M. B. Mannan, D. Barua, S. Islam, N. I. Khan and S. R. Mahmud, "Developing a Machine Learning Based Support System for Mitigating the Suppression Against Women and Children," *2021 5th International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*, Dhaka, Bangladesh, 2021, pp. 1-6, doi: 10.1109/ICEEICT53905.2021.9667924.

[6] Yang, M., Wong, S. C. P., & Coid, J. (2010). The efficacy of violence prediction: A meta-analytic comparison of nine risk assessment tools. Psychological Bulletin, 136(5), 740–767. https://doi.org/10.1037/a0020473

[7] P. Tamilarasi and R. U. Rani, "Diagnosis of Crime Rate against Women using k-fold Cross Validation through Machine Learning," 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2020, pp. 1034-1038, doi: 10.1109/ICCMC48092.2020.ICCMC-000193.

[8] A. Adhikary, S. A. Murad, M. S. Munir and C. S. Hong, "Edge Assisted Crime Prediction and Evaluation Framework for Machine Learning Algorithms," 2022 International Conference on Information Networking (ICOIN), Jeju-si, Korea, Republic of, 2022, pp. 417-422, doi: 10.1109/ICOIN53446.2022.9687156.

[9] Aziz, R.M., Sharma, P. & Hussain, A. Machine Learning Algorithms for Crime Prediction under Indian Penal Code. Ann. Data. Sci. (2022). https://doi.org/10.1007/s40745-022-00424-6

[10] J. Azeez and D. J. Aravindhar, "Hybrid approach to crime prediction using deep learning," *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Kochi, India, 2015, pp. 1701-1710, doi: 10.1109/ICACCI.2015.7275858.

[11] https://www.who.int/news-room/fact-sheets/detail/violence-against-women

[12] https://plan4womenssafety.dss.gov.au/resources/what-is-violence-against-women/

[13] https://www.clearias.com/violence-against-women/#:~:text=In%20respect%20of%20India,-According%20to%20the&text=Most%20of%20the%20cases%20of,and%20rape%20(7.4%20percent).

[14] https://www.bbc.com/news/world-asia-india-62830634

[15] https://evaw-global-database.unwomen.org/en/countries/asia/india