1. INTRODUCTION

1.1 Project Overview

In the judicial system, the scope of work with text documents is very significant, and the decision-making process must always be fair and transparent. Manually processing such a volume of information is very difficult and sometimes almost impossible. In addition, people without legal education and involved in the trial are faced with many problems and issues that are difficult to solve without asking a lawyer. Court judgments play a crucial role in litigation, legal study, and court decision-making because some of them are exemplars of legal usage and interpretation. Lawyers use these judgments to analyze if their clients could win the lawsuits.

With components of civil law, common law, equitable law, customary law, and religious law, India has a hybrid legal system. The Indian Penal Code (IPC), India's primary criminal code, serves as the foundation for this legal framework. It is a thorough code that aims to cover all important areas of criminal law. 23 chapters and 11 parts make up its divisions.

Automatic Charge Identification is the task of identifying the relevant legal charges given the facts of a situation and the statutory laws that define these charges and is a crucial aspect of the judicial process. This model focuses on Automatic Charge Identification and takes it a step further by matching these charges to the relevant IPC sections. Natural Language Processing and Deep Learning Methods are used to create this model.

This process is a crucial aspect of the judicial process and this model will be able to provide legal assistance to students, lawyers, prosecutors, judges, police as well as the general public. This project will also act as a useful tool in expediting legal cases through the use of technology, which will lead to justice being served more efficiently. As machine learning applied to the justice system is still a relatively unexplored concept, we believe that this project will act as a relevant step in this direction, and will present useful outcomes.

In the present scenario, the violence and the increasing crimes against women are witnessed by everyone across the world in some or other manner. It reflects the magnitude and

ponderousness of the atrocities committed against women in recent years. This is evidenced by the global crusade to eradicate violence against women. Hence for this project, we will be specifically focusing on sections highlighting Offences against Women in the Indian Penal Code, 1860 i.e., Section 302, 363, 366, 357 and 376. The aforementioned sections primarily focus on Murder, Kidnapping and Abduction of Women and Sexual Harassment and Rape.

1.2 Need Analysis

With the development and innovation of machine learning technology, more and more fields try to apply artificial intelligence to practical scenarios.

Legal Teams and prosecutors are always overwhelmed with multiple cases at a time which leads to delays in justice. We believe that this tool will act as an advantageous tool for those who lack resources and time, and would act as an assistant for law firms as well.

Lawspective is trained to automatically identify charges based on the facts of a situation, as well as match them to the relevant IPC sections at a human level without the limit of human capacity. This tool gives lawyers the time required for hyper-specialization and to improve their rapid expansion of discovery. This also helps lawyers to improve their level of communication and attention to their clients.

This tool can also be used by law students, which would give them more hands-on experience while still at university. These experiences would help in enriching their minds and retain more information.

This can also be used by the police which will lead to fewer unjustified arrests. Furthermore, the tool will be user user-friendly, making it convenient for the general public's usage as well.

This model will currently be Section 302, 363, 366, 357 and 376. The aforementioned sections primarily focus on Murder, Kidnapping and Abduction of Women and Sexual Harassment and Rape.

These sections focus on punishment against Murder, Kidnapping and Abduction of Women and punishment for Rape. These sections are quite diverse but have multiple crossovers in cases, making them ideal for multiple charge identification.

1.3 Research Gaps

The following were the drawbacks of the models researched:

- The performance of previous models can be improved with other variations of Machine learning. Also, utilization of more records in the dataset and building a model that can practically represent the sophisticated criminal law structure.

- Multi-Label Charges can be introduced by expanding more application scenarios, making the research more practical.

- Most models did not predict the IPC sections and instead only predicted broadly classified crimes.

- Most models worked more efficiently when they contained only one charge.

- Reinforcement learning can be introduced to generate latent fact details.

1.4 Problem Definition and Scope

Develop a machine learning model which would perform Automatic Charge Identification based on the facts of a situation, as well as match the identified charges to the relevant sections of the Indian Penal Code. The sections taken into consideration in this project are 375, 376, 302 and 363 and 366. These sections are based on Murder, Kidnapping and Abduction of Women and Punishment for Rape.

1.5 Assumptions and Constraints

Table 1: Assumptions and Constraints

S. No	Assumptions
1.	It is assumed that the end users of this product have a smartphone or a laptop/PC with an operating system or a browser.
2.	Deployment of this website will require proper licensing and code signing via appropriate authorities.
3.	Final implementation of this project will also require that the users understand their roles properly and how to use the website responsibly

1.6 Standards

Table 2: Standards

S.No	Standards	Details
1	**Web 2.0**	Web 2.0 refers to websites that emphasize user-generated content, ease of use, participatory culture, and interoperability for end users.
2	**ISO/IEC 90003**	Software engineering -- Guidelines for the application of ISO 9001:2008 to computer software is a guideline developed for organizations in the application of ISO 9001 to the acquisition, supply, development, operation, and maintenance of computer software and related support services.
3	**ISO/IEC/IEEE 29148**	This standard provides details for the processes and products related to the engineering of requirements for software products (including services) and systems throughout their life cycle. It defines the construct of a good requirement, provides attributes and characteristics of requirements, and discusses the iterative and recursive application of requirements processes throughout the life cycle. It also provides guidance in the 7 applications of requirements engineering and management processes for requirements-related activities in ISO/IEC/IEEE 12207 and ISO/IEC/IEEE 15288.

1.7 Approved Objectives

- To study the judicial system of India and the Indian Penal Code along with its sections in detail.
- To create a Machine-Learning model that can be used for Automatic Charge Identification based on the facts of a situation, and for the prediction of relevant IPC sections based on these identified charges.
- To develop an online platform for all users to access this model for their specific purposes.
- To test the Model on real-world legal cases and test the accuracy for the same.

1.8 Methodology

1.8.1 Communication and Planning

The scope narrative of this project was defined and the requirement analysis was carried out. This was followed by various interactive sessions with individuals who could provide assistance in a legal capacity and provide an adequate outline of the judicial system of India and the Indian Penal Code. This was followed by the development of a project plan.

1.8.2 Modelling

Based on the understanding of the project, Work Breakdown Structure, Design Level Diagrams, and User Interface Diagrams are created.

1.8.3 Construction

1.8.3.1 Data Collection

Collection of data manually through various internet sources using web scraping. The cases are collected in PDF format. Each case is about 15-20 pages long containing lots of metadata such as petitioner name, respondent name, date of judgment, name of judges involved in the case, date, place of occurrence of crime, IPC (Indian Penal Code) Sections under which case was registered and judgment. Most of the information was futile but relevant information was used later to carry out the analysis. For each section, 100 cases are collected, which makes a total of 350 cases as our dataset. Some of these cases have multiple sections applied to them.

1.8.3.2 Data Pre-processing

This textual data will be cleaned and formatted using various NLP methods. The dataset of these cases was built for evaluating our prediction model. With manual efforts and text summarization tools, information was extracted and short summaries were created from lengthy case descriptions, resulting in the dataset.

1.8.3.3 Model Creation

This model will be created and trained using various machine learning and deep learning models such as LSTM, GRU and BERT.

1.8.3.4 Software Product Development

It will include the development of a modular web application for delivering the product to various stakeholders. UI will be based on the idea that the interface is user-friendly for the general public.

1.8.3.5 Testing

Unit testing, Integration testing, and System testing like white and black box testing on the proposed system for obtaining results.

1.8.3.6 Deployment

The system will be deployed on the cloud.

1.9 Project Outcomes and Deliverables

This project will have the following deliverables: -

- A machine learning model for Automatic Charge Identification and matching the identified charges with the relevant sections of the Indian Penal Code.
- A digital platform for users to access this model in a user-friendly environment.

1.10 Novelty of Work

- There are very few Automatic Charge Identification models based on the Indian Criminal Code.
- This model also incorporates the element of IPC section matching, which has only been incorporated in a couple of previous models.

2. REQUIREMENT ANALYSIS

2.1 Literature Survey

2.1.1 Theory Associated with Problem Area

2.1.1.1 The Judicial System of India

The Indian judicial system follows the common law system based on recorded judicial precedents as inherited from the British colonial legacy. The court system of India comprises the Supreme Court of India, the High Courts and subordinate courts at district, municipal and village levels.

2.1.1.2 Indian Penal Code

The official criminal code of India is known as the Indian Penal Code (IPC). It is a thorough code that aims to cover all important areas of criminal law. The first law commission of India, headed by Thomas Babington Macaulay and constituted in 1834 as a result of the Charter Act of 1833, made suggestions that served as the basis for the creation of the code. It became operative in British India in 1862, at the beginning of the British Raj. However, it did not immediately apply in the Princely states, which until the 1940s had their own courts and legal systems. Since then, the Code has undergone numerous revisions and now includes more penal laws.

The IPC in its various sections defines specific crimes and provides punishment for them. It is subdivided into 23 chapters that comprise 511 sections.

2.1.1.3 Sections Focused on this project

- **Section 302: Punishment for Murder**: Anyone found guilty of murder faces the death penalty or a life sentence in prison, as well as a fine.
- **Section 363: Punishment for Kidnapping:** Anyone who kidnaps someone from India or from legal custody may be penalised with either type of imprisonment for a duration that may last up to seven years, as well as being subject to a fine.

- **Section 366**: **Kidnapping, abduction or inducing woman to compel her marriage, etc:** Anyone who kidnaps or abducts a woman with the intention of forcing her to marry someone against her will, or knowing it is likely that she will be forced to marry someone against her will, or with the intent of forcing or seducing her to engage in illicit sexual activity, or knowing it is likely that she will be forced or seduced to engage in illicit sexual activity, shall be punished with imprisonment of either description for a term that may extend to ten years, as well as being subject to a fine.
- **Section 375 : Rape**
- **Section 376: Punishment for Rape:** Unless the woman raped is his own wife and is not under the age of twelve, in which case he shall be punished with imprisonment of either description for a term which may extend to two years, with fine, or with both, and shall also be liable to fine. Rape is punishable by either description of imprisonment for a term which shall not be less than seven years but which may be for life.

2.1.2 Existing Systems and Solutions

- **Legal Judgement Prediction System [1]:** This project provides a model for legal proceedings, particularly murder cases, in India. Our model was trained using a variety of classical machine learning methods, including Naive Bayes, Random Forest, and SVM. When compared to the other two strategies, it was concluded that Naive Bayes delivered satisfactory results with the highest accuracy of 74%.

- **A Neural-Network-Based Model of Charge Prediction via the Judicial Interpretation of Crimes [3]**: The artificial intelligence (AI) based legal assistant system has made some progress with the neural-network-based charge prediction, which predicts the defendants' charges from the criminal case records using neural networks. This work focused on classifying unbalanced datasets and ambiguous crime interpretation as external knowledge was included to the classification to enhance charge prediction. The problem is that there are too many different criminal behaviours to accurately portray. Fortunately, the criminal behaviours may be linked to a few distinct charges, and each of these charges—referred to in

this work as prejudged charges—has a distinct crime interpretation in the legal system.

- **Automatic Charge Identification from Facts: A Few Sentence-Level Charge Annotation is All you Need [4]:** Automatic Charge Identification (ACI), which is the task of identifying the appropriate charges in light of the facts and the statutes that describe these charges, is a significant component of the legal process. Enhancing fact-side representations has not received much attention; instead, current efforts are focused on learning charge-side representations through modelling interactions between the charges. We have noted that just a small portion of the sentences in the facts actually connect to the charges. We show that by integrating a relatively small percentage (3%) of fact descriptions annotated with sentence-level charges, we can improve across a variety of different ACI models in comparison to modelling simply the principal document-level job on a noticeably larger dataset. Additionally, we propose a unique model that, within a multi-task learning framework, combines the major task of document-level charge recognition with a secondary duty of sentence-level charge labelling. Many current baselines for ACI are greatly outperformed by the suggested approach. The rare charges, which are notoriously challenging to identify, exhibit the biggest improvement in performance.

2.1.3 Research Findings for Existing Literature

Table 3 : Literature Survey

S. no	Roll Number	Name	Paper Title	Tools/ Technology	Findings	Citation
1	**101903142**	**Ayush**	IRJET-Legal Judgement Prediction System	They used a variety of conventional machine learning methods, including Naive Bayes, Random Forest, and SVM, to train their model. In	They used a machine learning algorithm to anticipate the potential IPC Section that will be applicable	**[1]**

				contrast to the other two strategies, they discovered that Naive Bayes offered satisfactory results with the highest accuracy of 74%. Additionally, they created Cases and Law databases, each of which contains 50 legal parts and 100 murder cases from lengthy cases that were in pdf format.	depending on the facts of the case after analysing the case's basic description. Additionally, information about fines, accusations, and legal restrictions will be provided. The results of the forecasts can assist judges and attorneys in their decision-making, but they can also aid non-legal professionals in their comprehension and assessment of the case. They will concentrate on enhancing the functionality of their model in the future using other kinds of machine learning. Additionally, they will think about including more records in their cases dataset and developing a model that can accurately depict the complex	

					criminal law system.	
2	**101903142**	**Ayush**	Predicting Outcomes of Court Judgments - A Machine Learning Approach	Logistic Regression (LR) classifier is used in the first technique, known as TFIDF-LR. The main difference between the second approach, TFIDF-SVM, and TFIDF-LR is that TFIDF-SVM employs linear SVM as opposed to an LR classifier. The third technique, CNN-W2V-EMB, combines Word2Vec embeddings with a Convolutional Neural Network (CNN) classifier. The Long Short-Term Memory (LSTM) Network model is implemented using Kera in the fourth technique, referred to as LSTM- W2V-EMB. A fifth approach, LSTM-Keras-EMB, is similar to LSTM-W2V-EMB but uses the regular Keras embedding layer rather than the unique Word2Vec embedding.	They took into account decision prediction, a crucial legal text analytics assignment for lawyers. They created and made a labelled, encoded database of 300 felony bail decisions available to the public. Future study may investigate hybrid neural network models. They intend to break longer sequences into smaller ones, experiment with them, add more features, POS tags, and legal references to their dataset. They also intend to use the same models for various kinds of court decisions.	**[2]**

3	**101903094**	**Anika**	A Neural-Network-Based Model of Charge Prediction via the Judicial Interpretation of Crimes	LSTM neural network was used in this project. 3 main function modules were used: Firstly, a Fact Encoder is used to generate vector representations of fact descriptions. Secondly, a Crime Interpretation Encoder deduces the interpretations of crimes highly related to factual description and converts them into vector representations. Finally, a Charge Predictor is used by which the vector representation of crime interpretation is integrated with the weighted factor vector to predict the final probability distribution. Three public datasets obtained from the Chinese Government were published on China Judgements Online.	The final prediction results are calculated on the vectors based on an LSTM neural network. Experiment results demonstrate that their model performed better than the baseline models on real-world datasets of different scales. In addition, they also validated the effectiveness of the acquisition of prejudged charges. Future Scope: Their method reflects that a case is not only related to a charge. multi-label charge prediction can be introduced, expand more application scenarios, and make the research more practical.	**[3]**
4.	**101903094**	**Anika**	Automatic Charge Identification from Facts:	In this project, LSTM neural network was utilised. There were	Their suggested model can accurately forecast all charges at the	**[4]**

			A Few Sentence-Level Charge Annotations is All You Need	three primary function modules: First, fact descriptions are converted into vector representations using a Fact Encoder. The second step is to translate the interpretations of crimes that are closely connected to factual descriptions into vector representations using a criminal interpretation encoder. In order to anticipate the final probability distribution, a charge predictor is utilised, which integrates the vector representation of crime interpretation with the weighted factor vector. China Judgements Online has made available three public datasets that were received from the Chinese government.	document level, but it is unable to distinguish between unlawful assembly and robbery for two of the penalties. Future Work: Rather than forecasting crimes (IPC Topics) as was done in this paper, they could focus on predicting the pertinent Sections in IPC directly.	
5	**101903018**	**Abhinav**	Verdict Prediction for Indian Courts Using Bag of	In order to classify each case into its charges (as per Indian judicial legislation), to determine whether	The Indian judicial system relied on the manual gathering and analysis of data or evidence for	**[5]**

			Words and Convolutional Neural Network	or not it is a bailable offence, and to provide an approximative judicial verdict, CNN employed the Bag of Words technique, which showed an average accuracy of 85% in prediction. The Courts of India's publicly available data, which included more than 200 criminal judicial records and 200 civil judicial records involving violations of 26 different counts under the constitution, was utilised.	judgments, but by implementing our CNN model, accurate and impartial judgments can be rendered in a significantly shorter amount of time. Future Aims: The model performs better with a single charge in the case. The ability to accurately estimate the verdict declines as the number of charges rises. They hope to create a model in the future that will function well with a larger variety of instances.	
6	**101903018**	**Abhinav**	Interpretable Charge Predictions for Criminal Cases: Learning to Generate Court Views from Fact Descriptions	Charge labels have also been added to the Seq2Seq model, which helps the derived rationale's charge discriminations. By creating a dataset from Chinese government websites, this model is tested on criminal cases from China.	Court View Gen is useful because it makes charge prediction systems easier to read, creates legal papers and court views automatically from fact descriptions. Future Potential: Latent fact details could be produced through reinforcement learning. Future cases will take into	**[6]**

					account more intricate ones with numerous defendants and charges.	
7	**102097023**	**Taranum**	Few-Shot Charge Prediction with Discriminative Legal Attributes	A few shot neural model that unifies the charge prediction task and the legal prediction task.	proposed a unique framework for multitasking learning to jointly infer a case's charges and attributes. To do this, an attribute attention technique was used. Improvements of more than 50% for light shot loads. Future Sight: It will be difficult to manage criminal trials with several defendants and charges, among other complications.	**[7]**
8	**102097023**	**Taranum**	CAIL2018: A Large-Scale Legal Dataset for Judgment Prediction	Charge labels have also been included to help the generated rationales' charge discriminations, which employs the Seq2Seq model. More than 2.6 million criminal cases were gathered for CAIL2018 from http://wenshu.court.gov.cn/.	Released the first comprehensive dataset for predicting legal judgments. More thorough annotations that are consistent with real-world settings are reserved for CAIL2018.	**[8]**

				5,730,302 criminal documents obtained from China Judgements Online were used to create CAIL2018.		
9	**101903094**	**Anika**	Learning to Predict Charges for Criminal Cases with Legal Basis	To create the fact embedding, the input fact description is fed into a document encoder, where global word-level and sentence-level context vectors are employed to carefully choose instructive words and sentences. China Judgements Online provided the dataset, and 50,000 of the papers were used for training, 5,000 for testing, and 5,000 for validation.	For charge prediction, it is required to take into account both the facts and pertinent legal provisions. Although civil law judgements are formed in accordance with statutory laws, enormous fact-charges may, in certain cases, capture the implicit reasoning of the court.	**[9]**

2.1.4 Problem Identified

There is a major lack of projects related to Automatic Charge Identification, and there are only a handful of projects which have taken IPC matching into consideration.

In the field of Artificial Intelligence, IPC prediction is a challenging task for the following reasons. First, although legal interpretation is based on logical deduction, it is far too complex to handcraft rules and to imitate such tasks with a computational model. Also, it is difficult to obtain the public dataset of Indian cases and judgments. Even if there is some source to retrieve the online text of judgements, that mainly provides for search purposes only.

2.1.5 Survey of Tools and Technologies

2.1.5.1 Machine Learning

- Languages: Python, MATLAB
- NumPy Stack – NumPy, Pandas, SciPy
- Machine Learning Libraries – Sci-Kit Learn, TensorFlow 2.0, Pytorch
- NLP Libraries – NLTK , SpaCy, Pytextrank
- Deep Learning – LSTM, TF-IDF, HAN

2.1.5.2 Software Development

- Web Frameworks (Backend)- Flask(Python), Django(Python), NodeJS(JavaScript), Spring Boot(Java)
- Web Frameworks(FrontEnd) - CSS (Bootstrap, Material UI), JavaScript (React.js, Vue.js, Angular.js)
- Database - SQL (SQLite, PostGress, Oracle DB), NOSQL(MongoDB)

2.2 Software Requirements Specification

2.2.1 Introduction

2.2.1.1 Purpose

The Purpose of Lawspective is to:

- Serve as a tool that can Automatically Identify legal charges based on the factual description of a situation.
- Match the identified charges to the relevant sections of the Indian Penal Code.

2.2.1.2 Intended Audience and Reading Suggestions

The Intended Audience for this tool consists of Lawyers, Judges, Public Prosecutors, Police, Law Students as well as the general Public. This tool acts as a legal assistant and is user-friendly, and can provide understanding and knowledge to everyone regardless of their legal knowledge.

2.2.1.3 Project Scope

This project can be treated as a standalone project which will be accessible to the general public. It can be used as a legal assistant and would be an excellent addition to any legal team for expediting cases.

2.2.2 Overall Description

2.2.2.1 Product Perspective

This work aims to create a Machine Learning Model which will be able to perform Automatic Charge Identification based on the facts of a situation and match the identified charges to the relevant IPC sections. Machine Learning and Artificial Intelligence in the legal world is an unexplored crossover and has the potential to reach great levels, and we believe that this project would be an excellent addition to that. Figure 1 shows the system block diagram for Lawspective.

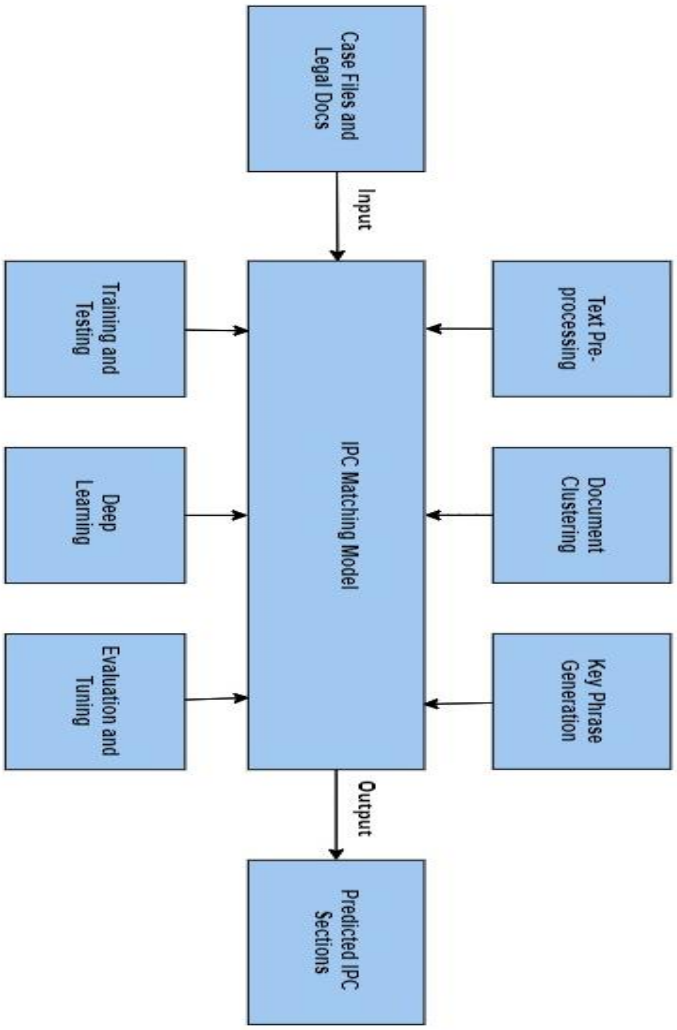

Figure 1: System Block Diagram

2.2.2.2 Product Features

- This system will act as a legal assistant for law firms, public prosecutors, judges, the police, law students and the general public.
- This system will Automatically Identify Legal Charges based on the facts of a situation.
- This system will match the identified charges to the relevant IPC sections.
- This system will provide information regarding the IPC sections predicted.

2.2.3 External Interface Requirements

2.2.3.1 User Interfaces

A graphical UI is created that provides a user-friendly environment for good visualizations. The user provides facts about the incident with useful keywords and then the interface links to the modules required for finding possible applicable charges' and further matching them with the IPC Sections. The results of matching can be seen on the web browser.

2.2.3.2 Hardware Interfaces

Not Applicable

2.2.3.3 Software Interfaces

In this system one of the major software interfaces is acting as a legal assistant by running models to identify legal charges and match them to relevant IPC sections, and displaying the results on the web browser. This acts as a software intermediary that allows users to interact with the model.

2.2.4 Other Non-Functional Requirements

The system is expected to meet the following Non-Functional requirements for a smooth and seamless experience with the system.

- **Safety**: Administrator should conduct a maintenance survey of the system after every six months.
- **Security**: The Admin should keep the server password secret and do not share it with anyone. Websites` should be secure according to the industry best practices.

- **Simple Interface**: Requirements for a UI ask that it be modern, easy to use and distinctive.
- **Scalable**: The website should be capable enough to handle 2000 users without affecting its performance.
- **Portable**: The software should be portable, moving from one OS to other OS does not create any problem. The website should be able to run across various platforms and screen sizes like mobile phones, tablets and full-size computers.

2.2.4.1 Performance Requirements

The performance of this tool is measured by how accurately it predicts the IPC sections and if it is able to identify the correct charges. If the legal charges are identified correctly, followed by the correct matching of the sections of the Indian Penal Code, only then the model has met all its performance requirements.

- Simplicity: For the UI, the main thing to keep in mind is to keep it simple and easy to use. It should not be uselessly sophisticated and complicated and usable for a layman. The user should only give input and get the results.
- Availability: The interface will be in use and accessible to the user till the internet is in working and that particular web page is open.

2.2.4.2 Safety Requirements

There is no such safety requirement but for a better experience administrator should conduct a maintenance survey of the system after every six months.

2.2.4.3 Security Requirements

The interface must be safe in terms of user data and confidentiality. Website would be secure according to the industry best practices and server password should remain a secret for the administrator.

2.3 Cost Analysis

The only cost required would be for website hosting as of now.

2.4 Risk Analysis

One of the major risks involving our project is the mismatching of IPC Sections, leading to the wrongful spread of information and can cause a problem with legal cases. This can improve over time as we train our model to predict sections more accurately.

3. METHODOLOGY ADOPTED

3.1 Investigative Techniques

Table 4: Investigative Techniques

S. no.	Investigative Project Techniques	Investigative Techniques Description	Investigative Projects Examples
1	Descriptive	Study of the problems faced during Automatic Charge Identification and IPC Matching. In the field of Artificial Intelligence, court judgement prediction is a challenging task for following reasons. First, even if logical reasoning is the foundation for legal interpretation, creating rules by hand or simulating such duties with a computational model would be far too difficult. Also, it is difficult to obtain the public dataset of Indian cases and judgements. Even if there is some source to retrieve the online text of judgements, that mainly provides for search purposes only. Therefore, some related work made their own	Projects: 1. Learning to How Does NLP Benefit Legal System: A Summary of Legal Artificial Intelligence. [13] 2. Predict Charges for Criminal Cases with Legal Basis, and How Does NLP Benefit Legal System: A Summary of Legal Artificial Intelligence [12] 3. Few-Shot Charge Prediction with Discriminative Legal Attributes. [13]

		dataset for model training and testing. Proper data analysis and model construction can avoid these problems very well. Therefore, tools to the intellectual analysis of the entire volume of information, to predict possible judicial decisions to citizens on the one hand, and to facilitate the routine work of lawyers on the other are required.	
2	Comparative	This model will be trained through various machine learning algorithms, which will be beneficial in comparing the accuracy and performance of the model based on the results of each algorithm. The machine learning algorithms which will be used are naïve bayes, random forest and SVM.	Projects: **1.** Legal Judgement Prediction System (IRJET) [1] **2.** CAIL2018: A Large-Scale Legal Dataset for Judgment Prediction [8]
3	Experimental	Several Classification Models and Charge prediction models will be employed as baselines. These will be experimented to one and based on that, the model with the best performance will be selected. The models	Projects : 1. Learn to Predict Charges for Criminal Cases for Legal Basis [9] 2. Interpretable Charge Predictions for Criminal Cases [6]

		that will be employed are TFIDF+SVM, CNN, LSTM, Fact Law Attention Model.	

3.2 Proposed Solution

This project aims to provide legal assistance through the development of machine learning tool which would be able to Automatically Identify Charges based on the facts of a situation and subsequently provide the relevant IPC sections based on the identified charges.

The proposed work is divided into two components:

1) Machine Learning
2) Software Development

3.2.1 Machine Learning

The Machine Learning solution involves 4 major steps to build a model Data Collection, Data Cleaning and Pre-Processing, Feature Extraction and Classification. This model will then be used to automatically identify charges and match them to relevant IPC sections.

- Data Collection: The dataset for this project is created manually by scraping text off various internet sources. This will include Case Files. In PDF format, the cases were compiled. Each case was between 15 and 20 pages long and contained a wealth of metadata, including the names of the petitioner and respondent, the date of the judgement, the names of the judges who were involved, the date, the location of the crime, the IPC (Indian Penal Code) Sections under which the case was registered, and the decision. Most of the data was useless, but later data that was useful was used to conduct the research.

- Data Cleaning and pre-processing: Applying Data Cleaning and pre-processing techniques to clean and structure the data. NLP techniques of removal of special symbols and stop words, and lemmatization and tokenization are done.

- Classification Algorithms: The plan is to apply various deep learning algorithms such as GRU, LSTM and BERT.

- Prediction: Using the Trained model to Charges and IPC sections.

3.2.2 Software Development

Software development includes the building of a modular web application for delivering the product to various stakeholders i.e. Law Firms, Public Prosecutors, The Government, Law Schools and for general public usage as well. Users would be provided with various questions where they describe the facts of their situations. The design component will heavily focus on making the system easy to use and provide simple understanding to users without overwhelming them with extensive legal knowledge that isn't understandable to common public.

The users will be able to view the identified charges, Relevant IPC sections and subsections. These sections will also consist of a concise explanation of the charges.

3.3 Work Breakdown Structure

Work can be made more manageable and approachable by using a common productivity strategy called task breaking. The Work Breakdown Structure (WBS), one of the most significant project management papers, is the tool that applies this technique to projects. It does it on its own, integrating scope, cost, and schedule baselines to guarantee project plans are in sync. Figure 2 refers to the work breakdown structure of Lawspective and it divided into 5 parts:

- Project Management
- Dataset Management
- Technical Designing
- Training and Testing
- User Interface

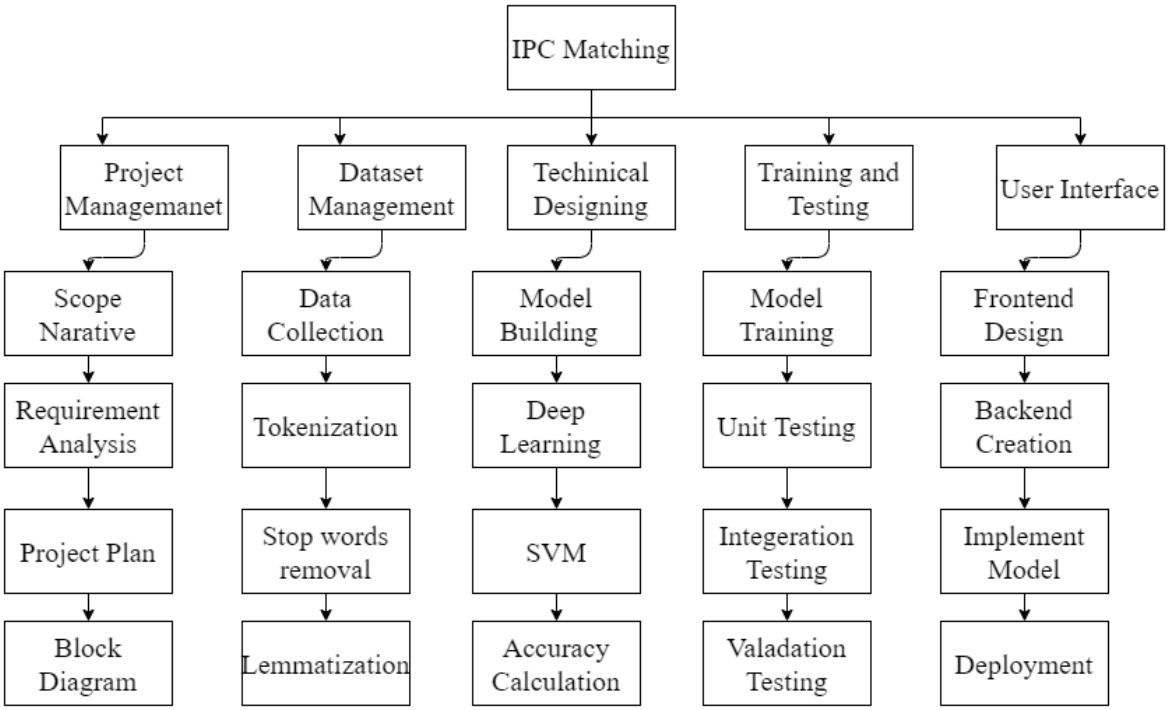

Figure 2 : Work Breakdown Structure

3.4 Tools and Technology

3.4.1 Machine Learning

- NumPy Stack including NumPy, Pandas, SciPy, Matplotlib
- Sci-Kit Learn
- TensorFlow 2/0
- Natural Language Toolkit
- Keras
- SpaCy
- Woosh

3.4.2 Web Development

- Web Frameworks (Backend)- Flask(Python), Django(Python), NodeJS(JavaScript), Spring
- Boot(Java) Web Frameworks(FrontEnd) - CSS (Bootstrap, Material UI), JavaScript (React.js, Vue.js, Angular.js)
- Database - SQL (SQLite, PostGress, Oracle DB), NOSQL(MongoDB)

3.4.3 Mock-ups and UX Design

- Draw.io
- Moqups.com

4. DESIGN SPECIFICATIONS

4.1 System Architecture

An application is divided into three primary logical components using the Model-View-Controller (MVC) architectural pattern: the model, the view, and the controller. Each of these parts is designed to handle particular application development facets. One of the most popular and widely accepted web development frameworks for building scalable and flexible projects is MVC. The MVC architecture is depicted below in Figure 3.

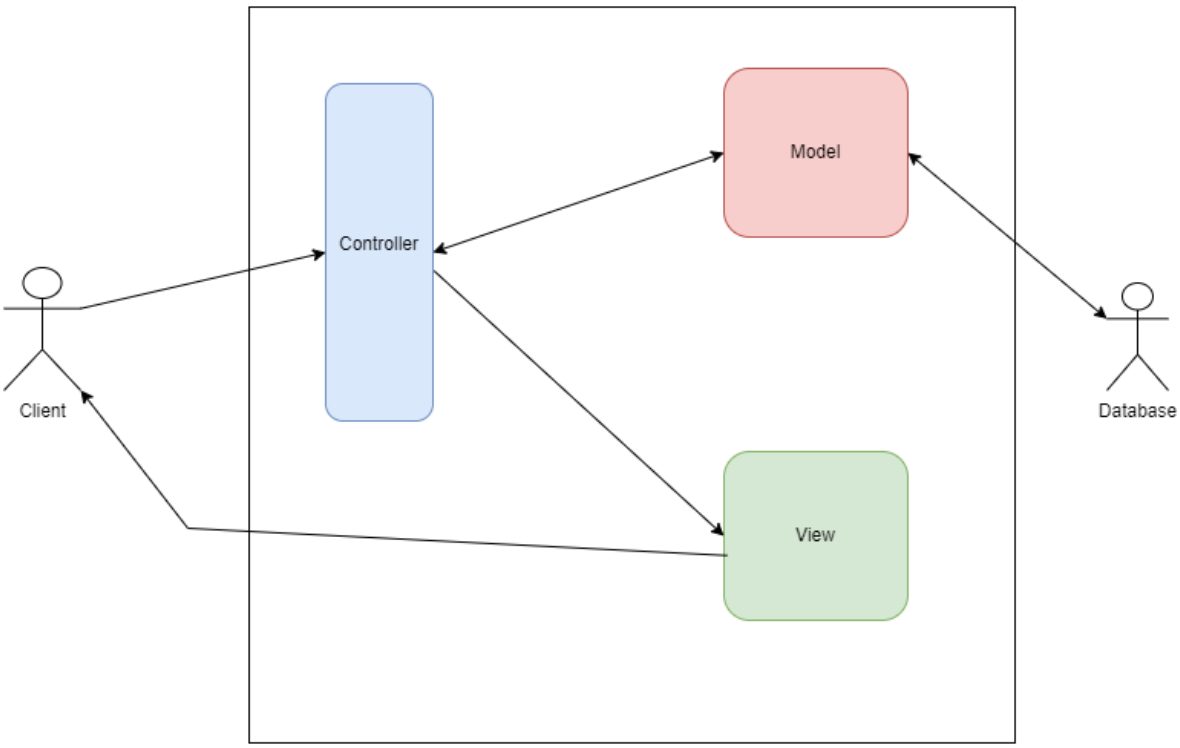

Figure 3 : MVC Architecture

4.2 Design Level Diagrams

4.2.1 Data Flow Diagrams

4.2.1.1 DFD LEVEL 0

Context Level Diagram at DFD Level 0. It is an introduction to the Lawspective and the entities involved. The system is displayed as a single high-level process, together with its relationship to external entities, in this quick-glance view. It ought to be simple enough for developers, business analysts, and data analysts to understand. Figure 4 refers to the DFD Level 0 diagram for Lawspective.

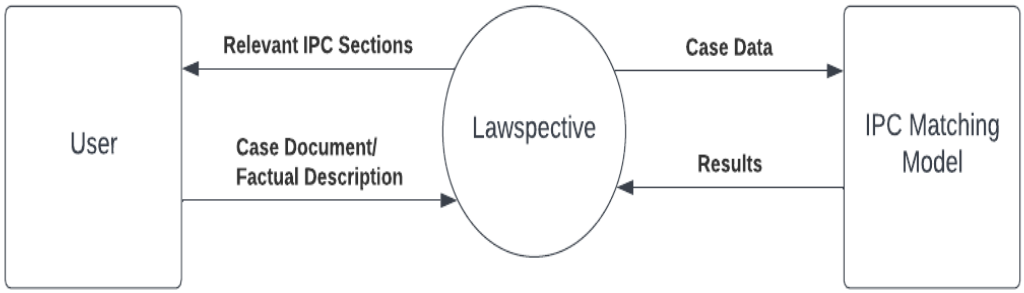

Figure 4: DFD Level 0

4.2.1.2 DFD LEVEL 1

The Level-0 Diagram is broken down in more detail in DFD Level 1. The Lawspective's primary responsibilities are highlighted, and each process is then broken down into further sub-processes with a thorough data flow to help the reader understand how each one operates. Figure 5 refers to level 1 Data Flow Diagram of Lawspective.

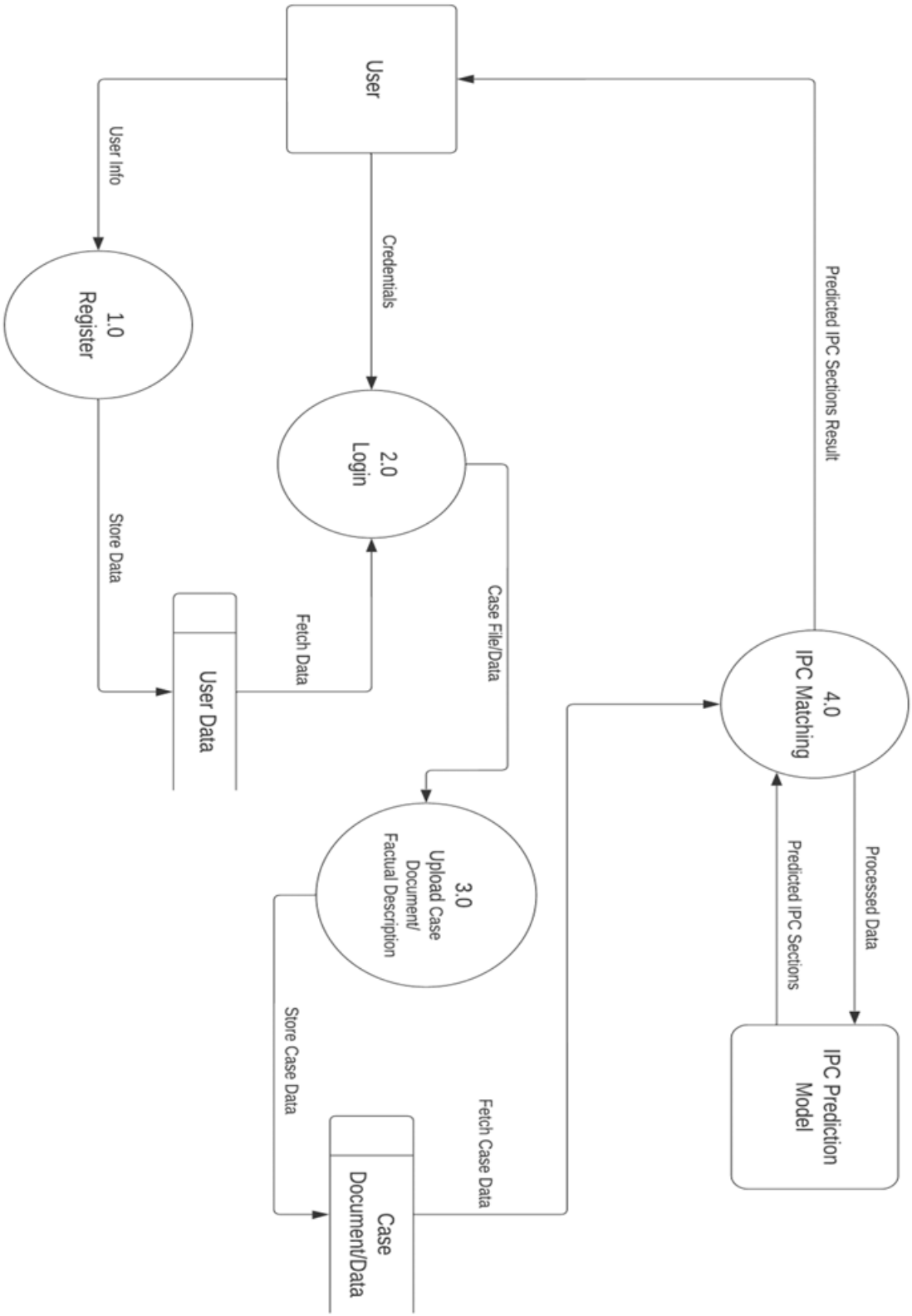

Figure 5: DFD LEVEL 1

4.2.2 Entity-Relationship Diagram

Interrelated topics of interest in a certain field of knowledge are described by the ER model for IPC matching model. It also demonstrates the connections between many classes, like user, admin, login, signup, etc., and their associated characteristics. Figure 5 refers to the ER diagram of Lawspective.

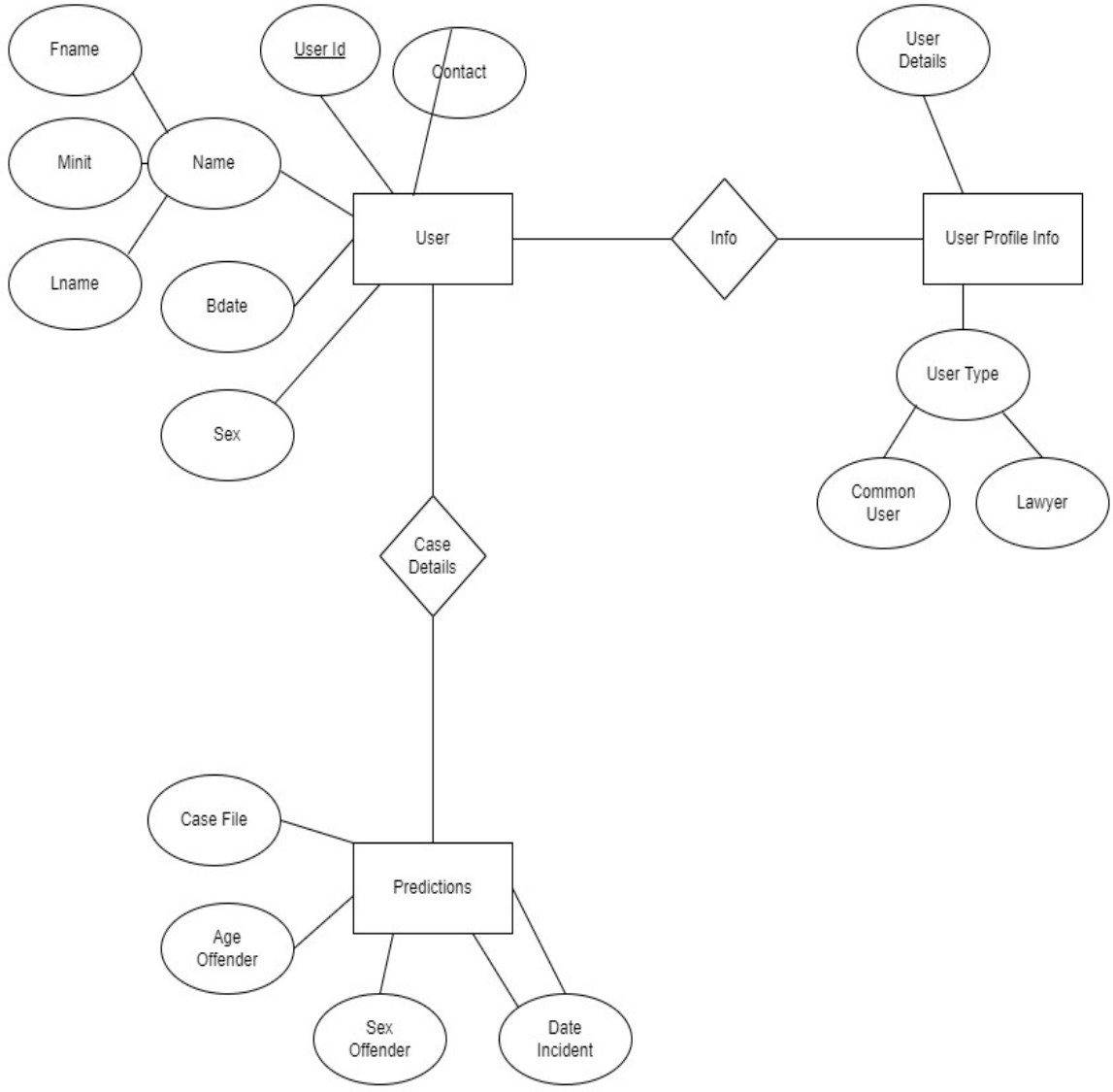

Figure : Entity Relationship Diagram

4.2.3 Class Diagram

The structure and characteristics of how our system functions are mapped out in this class diagram for the IPC matching model. Additionally, it explains the functionalities carried out by several classes, such as user, admin, login, signup, etc., and illustrates the relationships between them. Figure 7 refers to the Class Diagram of the project.

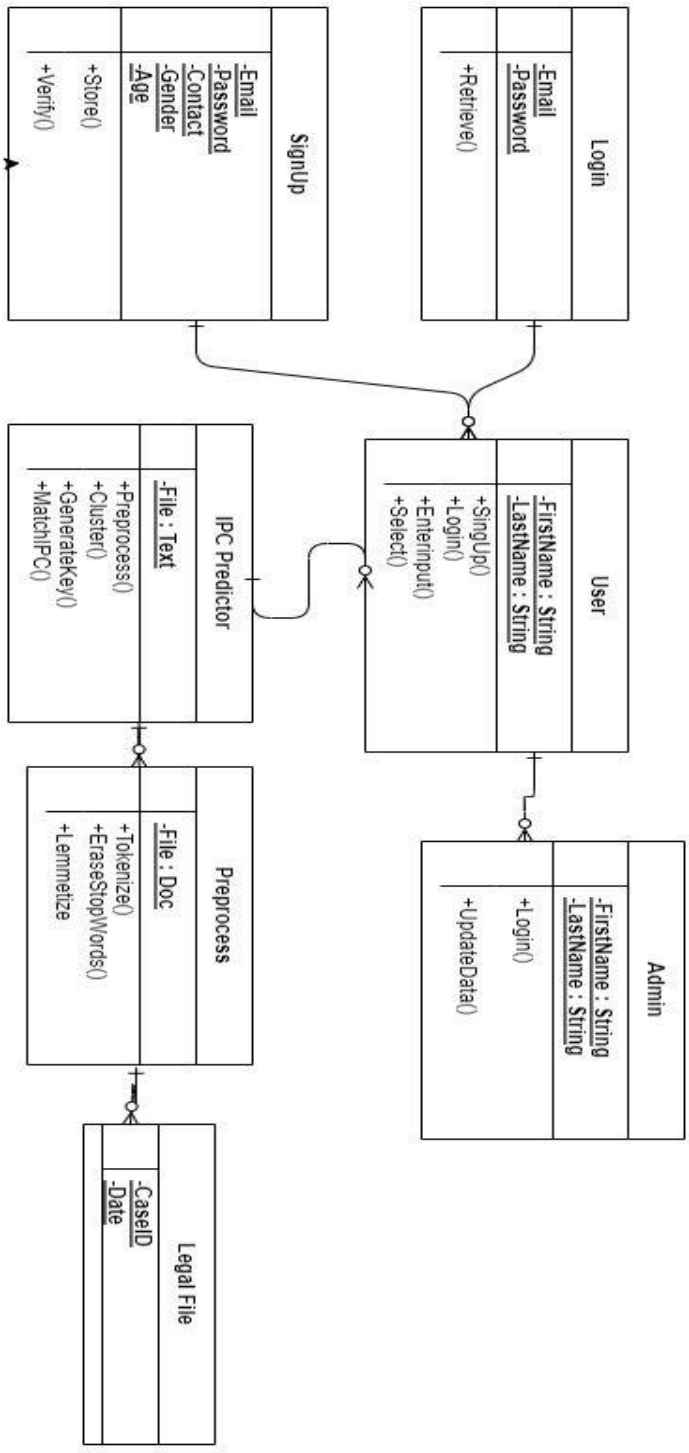

Figure 7: Class Diagram

4.3 User Interface Diagrams

4.3.1 Use Case Diagram

Figure 8 refers to the use case diagram of the project. A graphical representation of a user's potential interactions with a system is called a use case diagram. A use case diagram, which is frequently complemented by other types of diagrams, displays the numerous use cases and user types the system has.

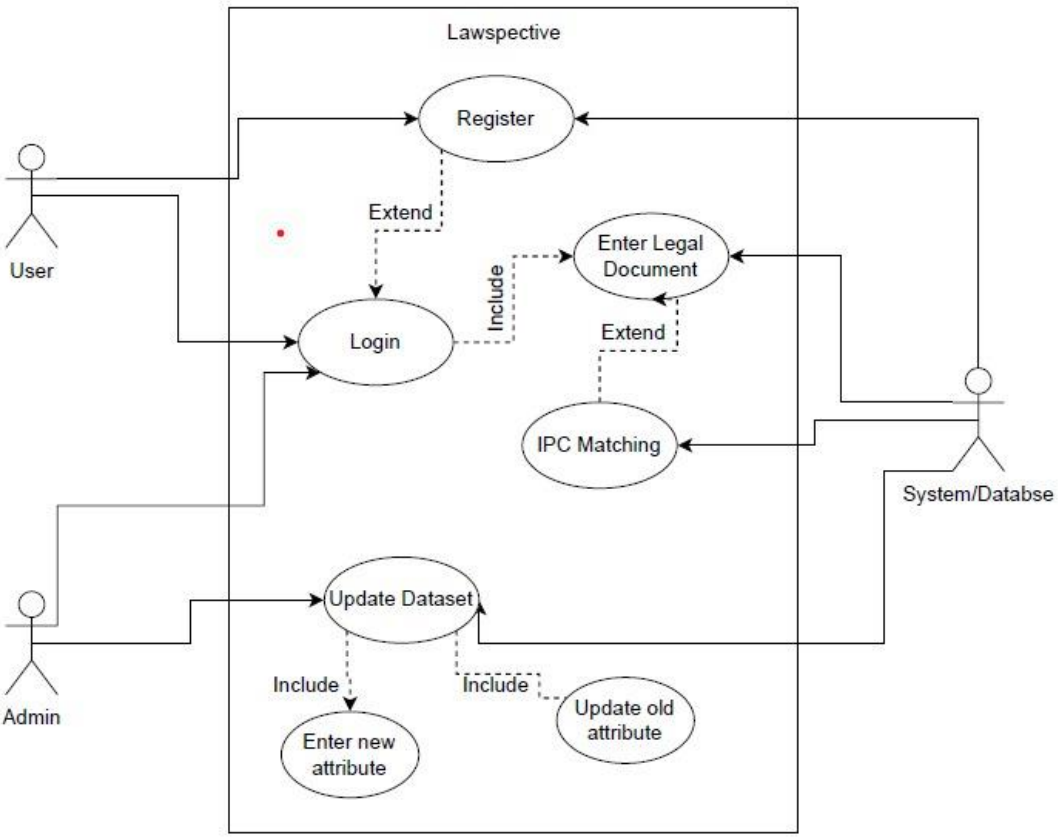

Figure 8: Use Case Diagram

Table 5 : Use Case Table

Use Case	IPC Sections Prediction system based on facts.
Use Case Purpose	To build a system for detection of IPC Sections that could be applicable based on the facts provided by the victim.
Use Case Description	Our system uses charge identification for the prediction of IPC Sections. This helps the user to know more about legal ways and the system. Also, it will help in the automaton of the judicial hearing in the future.
Pre-Conditions	Text provided to the system is correct and contains actual facts. Facts are related to the IPC Sections for women harassment.
Variations	Full legal documents may be provided in pdf format. Various deep learning algorithms can be used for prediction of IPC Sections.
Normal Scenario	1. Internet is connected 2. Text containing facts is fed as input 3. Text is pre-processed 4. Charges and IPC Sections are predicted based on facts 5. Accuracy is predicted and displayed

Extension Points	If text doesn't contain facts a message is shown.
Alternate Scenario	Case not related to IPC 1. Facts related to other cases than IPC is fed as input 2. Ensures that facts provided is related to IPC. 3. Return to normal scenario
Post Conditions	Post Condition: Success End Condition 1. IPC Sections are detected and displayed in real time 2. Insufficient description alert shown if accuracy is low Failure End Conditions 1. IPC Sections not detected 2. Incorrect data displayed

4.3.2 Activity Diagram

Another crucial behavioural diagram in the UML used to depict the system's dynamic elements is the activity diagram. An activity diagram is essentially a more complex flow chart that models the transition between activities. Figure 9 refers to the activity diagram of Lawspective.

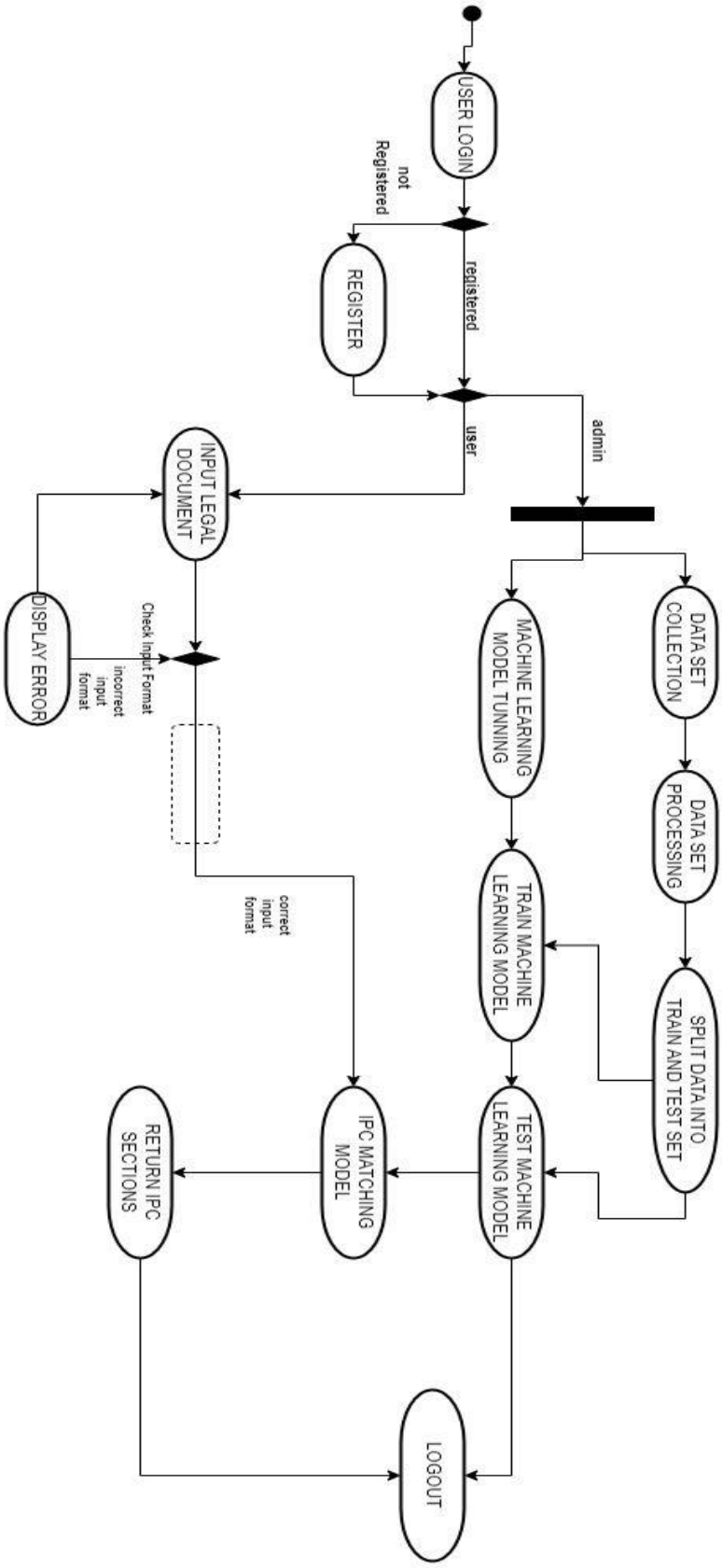

FIGURE 9: Activity Diagram

5. IMPLEMENTATION AND EXPERIMENTAL RESULTS

5.1 Experimental Setup

The official website of the Supreme court has been used for data collection and product testing.

Intel i5 processor with Windows 10/MACOS Operating System and 8GB RAM has been used for experimental work pertaining to Machine Learning as well as product development. Conda Distribution of Python 3.7 with the latest versions of Tensorflow-2, SciKit-Learn, NLTK, and other minor libraries were used for implementing and evaluating experiments pertaining to machine learning. For Product Development, Jetbrain's Pycharm IDE for Python Programming Language was used as the primary environment.

5.2 Experimental Analysis

5.2.1 Data

The Supreme Court of India is the supreme judicial authority of India and is the highest court of the Republic of India under the constitution. It is the most senior constitutional court and has the last say in all cases—aside from those involving personal laws—as well as the authority to conduct judicial reviews. The official website of the supreme court was used for data collection and product testing. The model was trained on a dataset of 300 cases from 2002 to 2022 on sections 302, 363, 366, 375 and 376, with 100 cases for each section. A few cases had multiple sections, which would be helpful in training our model for multiple charge identification. The data has been collected with keywords `rape', 'harassment', `kidnapping', 'abduction' and 'murder'.

The information extracted from these case files was manually cleaned and summarised. By creating the short summaries from lengthy pdf judgements, main attributes were extracted that would be sufficient to provide short description of any murder case. These attributes were facts, charges, and sections.

5.2.2 Performance Parameters

To evaluate the machine learning models, we used classification accuracy, precision, recall and F1-Score. Classification accuracy is the ratio of number of correct predictions to total predictions. Recall is the number of relevant documents retrieved divided by the total number of existing applicable documents, while precision is the number of relevant documents retrieved divided by the total number of documents retrieved. F1-Score is a metric which combines precision and recall as their harmonic mean. It is important because in machine learning problems, it has been observed that there is a trade-off between precision and recall, thus needing a simpler metric to get balance of both. These metrics are calculated as:

- **Accuracy** represents the number of correctly classified data instances over the total number of data instances.
- **Precision** is how good the model is at predicting a specific category.
- **Recall** tells you how many times the model was able to detect a specific category.
- **F1 score** is a machine learning evaluation metric that measures a model's accuracy. It combines the precision and recall scores of a model.

Table 6: Metrics

Metric	Formula
Accuracy	(tp+tn)/(tp+tn+fp+fn)
Precision	tp(tp+fp)
Recall	tp / (tp+fn)
F1 Score	2*R*P/(R+P)

5.3 Working of the Project

5.3.1 Procedural Workflow

- The process starts with a collection of cases via the official website of Supreme Court. The cases were collected based on keywords. The sections considered

were 302, 366 and 376, and keywords used are ‘rape’, ‘sexual harassment’, ‘kidnapping’ ‘abduction’ and ‘murder’.

- These case files are pr e-processed and facts are manually extracted and fed into machine learning models as a training set.
- This model will automatically identify charges based on the facts of a situation using the model.
- Based on the identified charges, the model predicts the relevant IPC sections. These can be one or multiple sections based on the number and variety of charges.
- These predictions help the user get an idea about the direction of the case and act as legal assistance.

5.3.2 Algorithmic Approaches Used

Various Machine Learning and Deep Learning Algorithms were considered for the training of classification models. The major ones which proceeded for comparative analysis were GloVe GRU, GloVe LSTM and BERT. After training and testing it was found that BERT was found to be superior for our application.

Word Embeddings

Word embeddings are a sort of word representation that allows for the depiction of words with comparable meanings. They are a distributed representation of text that may be one of the major innovations behind deep learning techniques' excellent performance on difficult natural language processing challenges. Three techniques can be used to learn a word embedding from text data.:

- **Embedding Layer**

  For want of a better term, an embedding layer is a word embedding that is learned along with a neural network model for a particular goal in natural language processing, such as document categorization or language modelling. It necessitates the cleaning and preparation of document text such that each word is one-hot encoded. The model includes a specification of the vector space's dimensions, which can be 50, 100, or 300. Small random numbers are

used to initialise the vectors. The Backpropagation algorithm is used to fit the embedding layer in a supervised manner on the front end of a neural network.

- **Word2Vec**

  A statistical technique called Word2Vec can effectively be used to learn a solitary word embedding from a text corpus. It was created by Tomas Mikolov and colleagues at Google in 2013 in an effort to improve the effectiveness of embedding training using neural networks, and it has now taken over as the industry norm. The effort also included investigating how vector algebra applied to word representations and analysing the learned vectors. For instance, the phrase "king is to queen as man is to woman" can be expressed by taking the "man-ness" out of "King" and adding the "women-ness."

- **GloVe**

  The Global Vectors for Word Representation, or GloVe, algorithm is an extension to the word2vec method for efficiently learning word vectors. Latent Semantic Analysis (LSA), a matrix factorization technique, was used to create traditional vector space model representations of words. While LSA is effective at exploiting global text statistics, word2vec is better at extracting meaning and exhibiting it on tasks like calculating analogies (e.g. the King and Queen example above). A method called GloVe combines the local context-based learning in word2vec with the global statistics of matrix factorization techniques like LSA. GloVe creates an explicit word-context or word context in place of using a window to define local context.

BERT

Bidirectional Encoder Representations from Transformers, or BERT, is a large neural network architecture with an enormous amount of parameters—up to 300 million in total. Consequently, overfitting would occur if a BERT model were trained from start on a limited dataset.

As a result, it is preferable to employ a pre-trained BERT model that was trained on a sizable dataset. The model can then be refined by running additional training on our comparatively smaller sample.

BERT is dependent on a Transformer (the attention mechanism that learns contextual relationships between words in a text). An encoder to read the text input and a decoder to create a prediction for the task make up a basic Transformer. Since the objective of BERT is to produce a language representation model, just the encoder portion is required. A series of tokens that are first transformed into vectors and then processed by the neural network make up the input to the BERT encoder. However, BERT requires the input to be modified and embellished with additional metadata before processing can begin:

- Token Embeddings: At the start of the first sentence, a [CLS] token is added to the input word tokens, and at the conclusion of each sentence, a [SEP] token is added.
- Segment embeddings: Each token has a marker that designates either Sentence A or Sentence B. Because of this, the encoder can tell which sentences are which.
- Positional embeddings: Each token is given a positional embedding to show where it belongs in the sentence.

8 layers have been used for this project: There are 3 input layers, 1 Keras Layer, 2 Dense Layers and 1 Dropout Layer.

Layer (type)	Output Shape	Param #	Connected to
input_word_ids (InputLayer)	[(None, 171)]	0	[]
input_mask (InputLayer)	[(None, 171)]	0	[]
segment_ids (InputLayer)	[(None, 171)]	0	[]
keras_layer (KerasLayer)	[(None, 768), (None, 171, 768)]	109482241	['input_word_ids[0][0]', 'input_mask[0][0]', 'segment_ids[0][0]']
tf.__operators__.getitem (SlicingOpLambda)	(None, 768)	0	['keras_layer[0][1]']
dense (Dense)	(None, 64)	49216	['tf.__operators__.getitem[0][0]']
dropout (Dropout)	(None, 64)	0	['dense[0][0]']
dense_1 (Dense)	(None, 3)	195	['dropout[0][0]']

Figure 10: Layers of Bert Model

5.3.3 System Screenshots

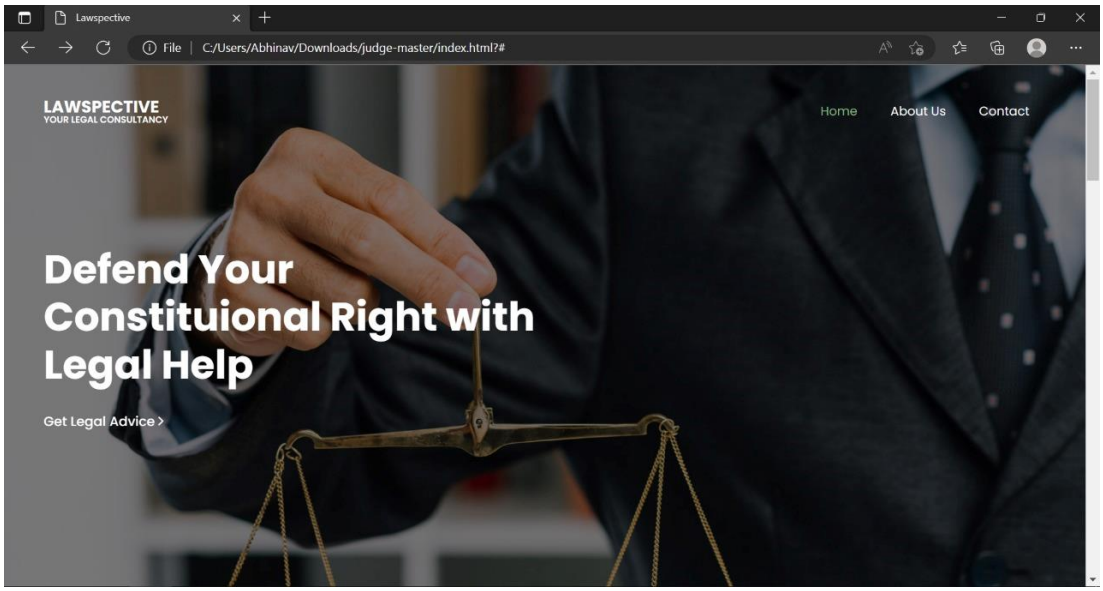

Figure 11: System Screenshots

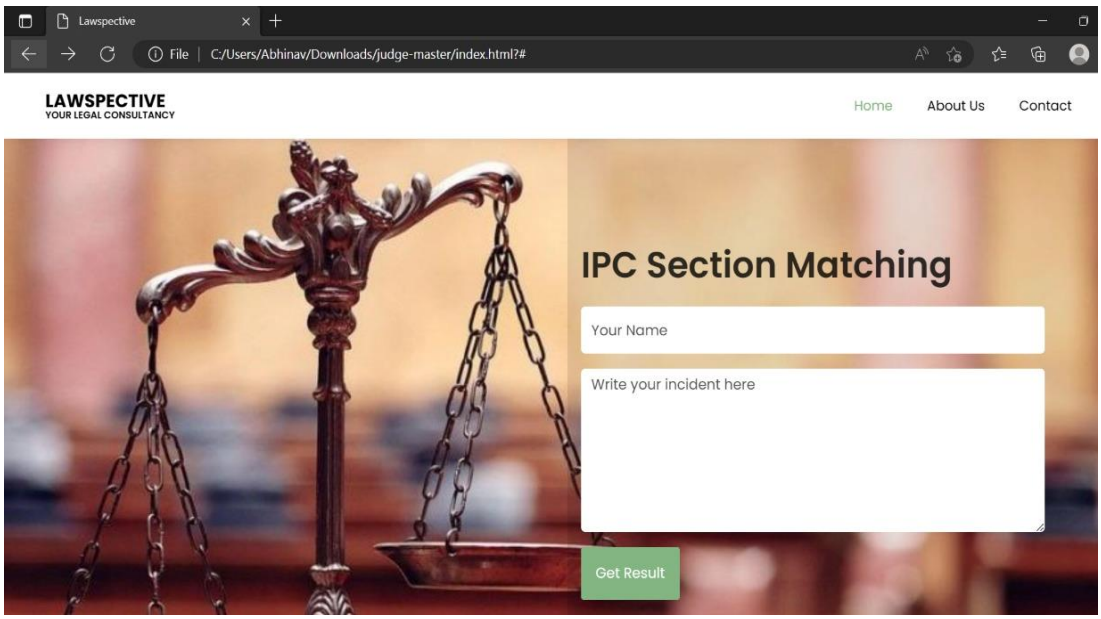

Figure 12: Input Page

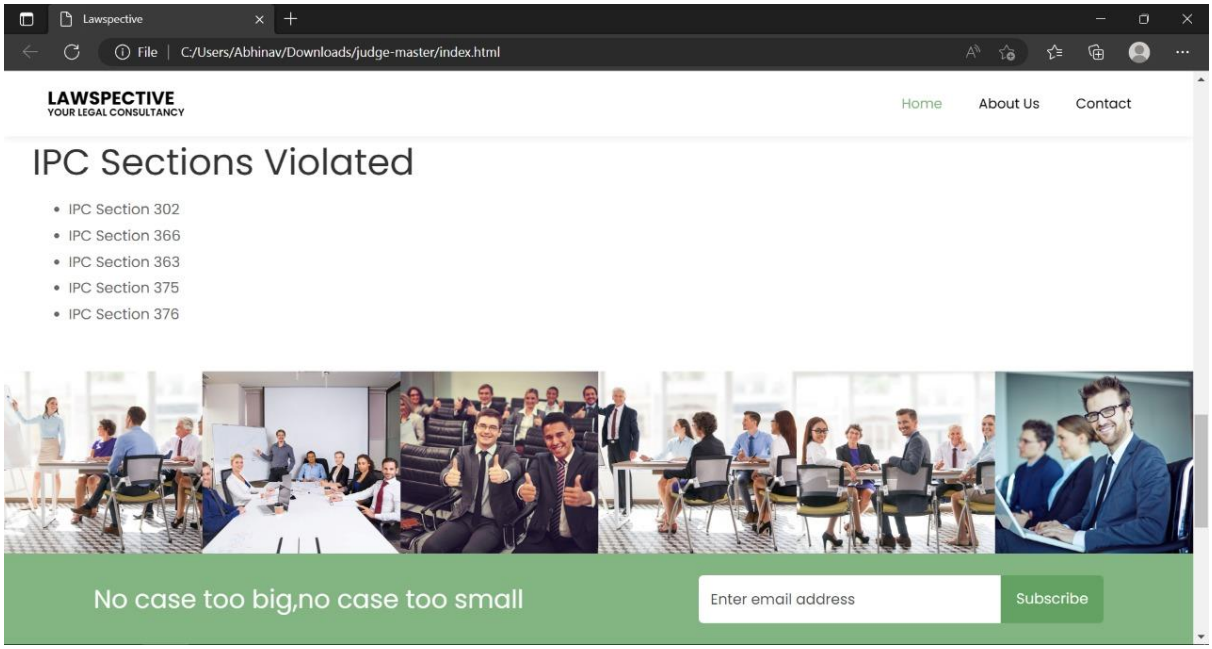

Figure 13: Output Page

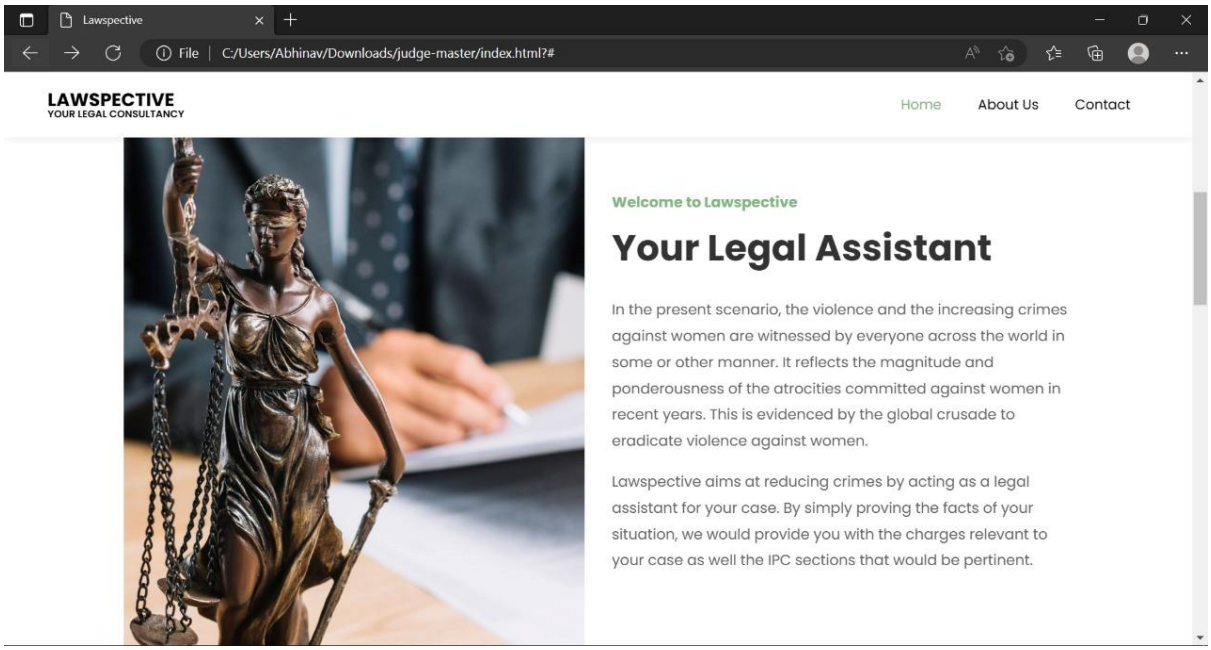

Figure 14: System Screenshots

5.4 Testing Process

This section describes the complete testing process including the techniques employed for testing and fixing bugs, test cases and results after testing.

5.4.1 Test Plan

The main idea behind the test plan consists of testing our proposed model with different hyper-parameters. The testing of the model was done on the test dataset from cases taken from the supreme court website. Different test cases were recorded based on the features of the applications provided, which are listed below in Table 7:

Table 7: Test Cases

S,No	Test Case	Input	Expected Output	Actual Output	Result
1	Acceptance of Name of user and text from case file in UI.	Name of user and text	Successful	Successful submission	Sucessful
2	Prediction of single charge and single section	Text from case file	Single IPC section	Successful IPC section predicted	Successful
3	Prediction of two charges and associated sections	Text from case file	Two IPC sections	Successful prediction of two IPC sections	Successful
4	Prediction of three charges	Text from case file	Three IPC sections	Successful prediction of three	Successful

	and associated sections			IPC sections	

5.4.2 Features to be Tested

- Classification Accuracy
- Precision
- Recall
- F1 Score
- Validation loss

5.4.3 Test Strategy and Techniques

The purpose of the model is to automatically identify legal charges based on the facts of the situation and match them to the relevant IPC sections. The testing is done based on classification accuracy, precision, recall, f1 score and validation loss.

5.4.4 Test Cases

The following were the important test cases to properly gauge the working of the software:

- Performance: performance was tested under different hyperparameters. Overfitting and underfitting were removed during the process.
- Outputs under different circumstances: The model will be tested under different cases with a single charge or multiple charges.
- Dataset Variances: Testing of the model using the dataset collected from the Supreme Court Website.

5.4.5 Test Results

- After repeated efforts in designing and implementing Bidirectional Encoder Representations from Transformers – BERT, a final optimized model was created with minimal possible overfitting and underfitting.

- Multiple cases with different legal charges and various numbers of legal charges were tested exhaustively.
- The model is able to predict charges accurately. However, as the number of charges increase, there is a decrease in the accuracy of the model.
- All features of the Lawspective website were found to be working properly.

5.5 Results and Discussion

Three models were trained and tested on our dataset of 350 cases.

Table 8: Results

S.no	Model	Accuracy	F1 Score	Recall	Precision	Loss
1	GRU	0.84	0.84	0.84	0.85	0.65
2	LSTM	0.77	0.77	0.77	0.78	0.84
3	**BERT**	**0.92**	**0.93**	**0.93**	**0.93**	**0.27**

BERT performs best as from the three models trained and tested on our dataset. Along with the highest accuracy, it also has the lowest loss.

The section wise precision, recall and f1-score can be seen in the figure below.

```
3/3 [==============================] - 1s 377ms/step
              precision    recall  f1-score   support

           0       0.93      0.93      0.93        30
           1       1.00      0.70      0.82        10
           2       0.91      1.00      0.95        30

    accuracy                           0.93        70
   macro avg       0.95      0.88      0.90        70
weighted avg       0.93      0.93      0.93        70
```

Figure 15: Metrics of BERT model

Here 0 represents murder charges, 1 represents kidnapping and abduction and 2 represents sexual harassment and rape. This further displays the relevant IPC sections for these charges.

Learning Curves

For algorithms that learn (optimise their internal parameters) incrementally over time, such as deep learning neural networks, learning curves are widely used.

The learning metric could be maximising, which means that higher scores (larger numbers) indicate more learning. One example is classification accuracy.

It is more common to use a minimising score, such as loss or error, where better scores (smaller numbers) indicate more learning and a value of 0.0 indicates that the training dataset was perfectly learned with no mistakes.

The current state of a machine learning model at each step of the training algorithm can be evaluated during training. It can be tested against the training dataset to determine how well the model is "learning." It can also be tested on a separate validation dataset that is not included in the training dataset. The validation dataset is used to determine how well the model "generalises."

- Train Learning Curve: A learning curve derived from the training dataset that indicates how well the model is learning.
- Validation Learning Curve: A learning curve derived from a hold-out validation dataset that shows how well the model generalises.

Dual learning curves for a machine learning model are commonly created during training on both the training and validation datasets.

In other circumstances, creating learning curves for several metrics is also popular, such as in classification predictive modelling challenges, where the model may be improved using cross-entropy loss while model performance is measured using classification accuracy. In

this example, two plots are generated, one for each metric's learning curves, and each plot can display two learning curves, one for each of the train and validation datasets.

- Optimization Learning Curves: Learning curves determined on the metric used to optimise the model's parameters, such as loss.
- Performance Learning Curves: Learning curves calculated on the measure that will be used to evaluate and select the model, such as accuracy.

Figure 16 represents the training and validation accuracy curve for BERT.

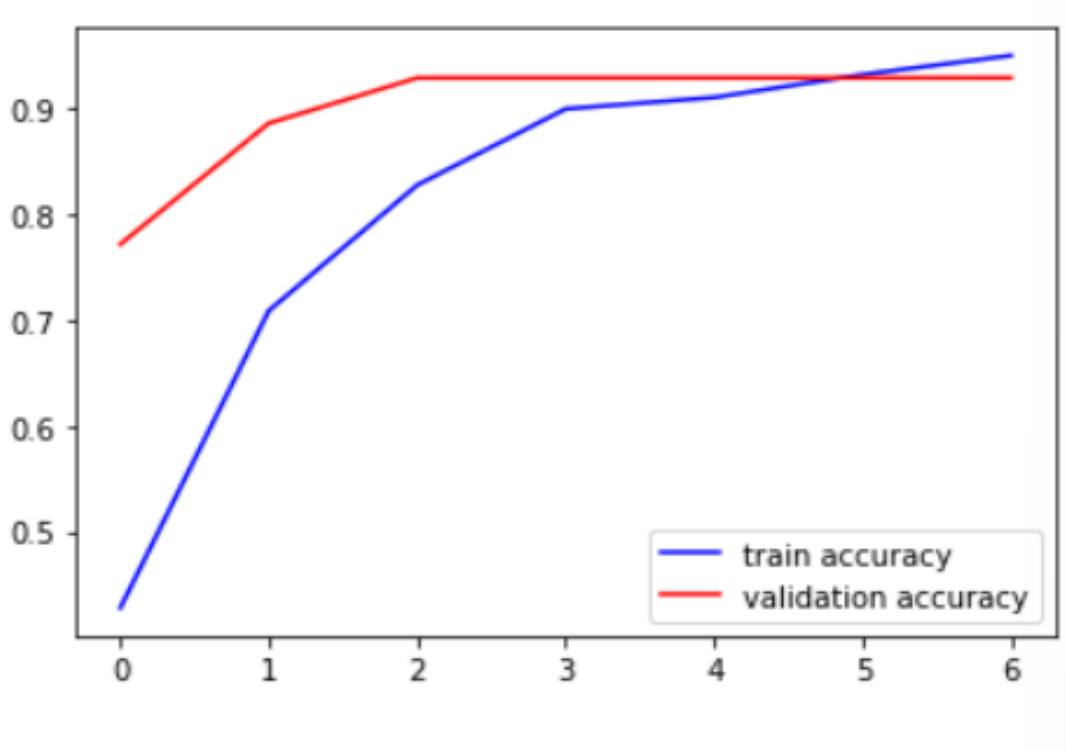

Figure 16: Training and Validation Accuracy Curve for BERT

Figure 17 refers to the training and validation loss curve for BERT.

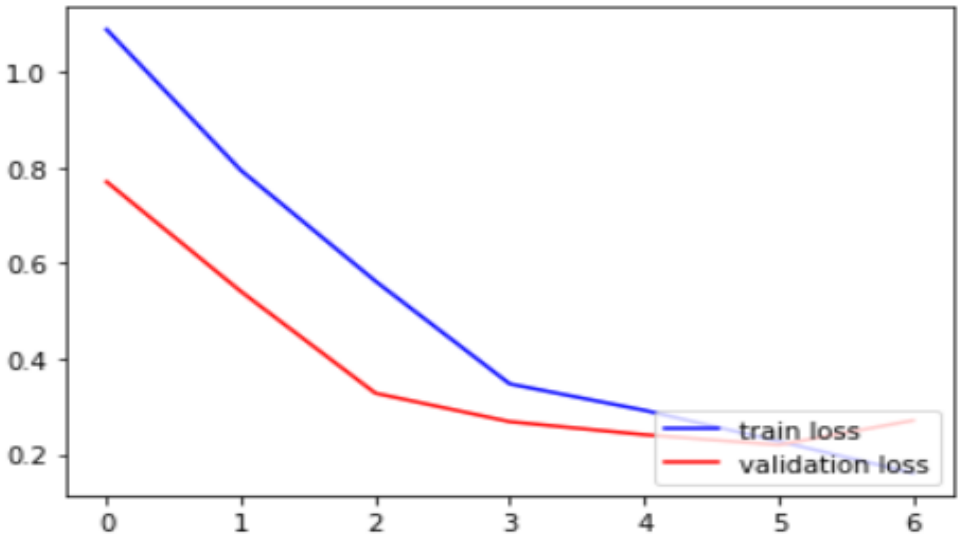

Figure 17: Training and Validation Loss Curve for BERT

Training Loss is the indicator of how well a deep learning model fits the training data. In other words, it evaluates the model's error on the training set. On the other hand, a deep learning model's performance on the validation set is evaluated using a statistic called validation loss. The dataset's validation set is a section set aside to check the model's efficacy. In the graph above, the training loss and validation loss both decrease and stabilize at a specific point. This indicates a good fit. Hence Bert is the perfect model for our problem statement.

When training loss and validation loss are both high, this may indicate that the model is underfitting. Underfitting occurs when the model is unable to accurately model the training data, and hence generates large errors. This can be seen in the training and validation loss graph for our GloVe LSTM model below in Figure 13.

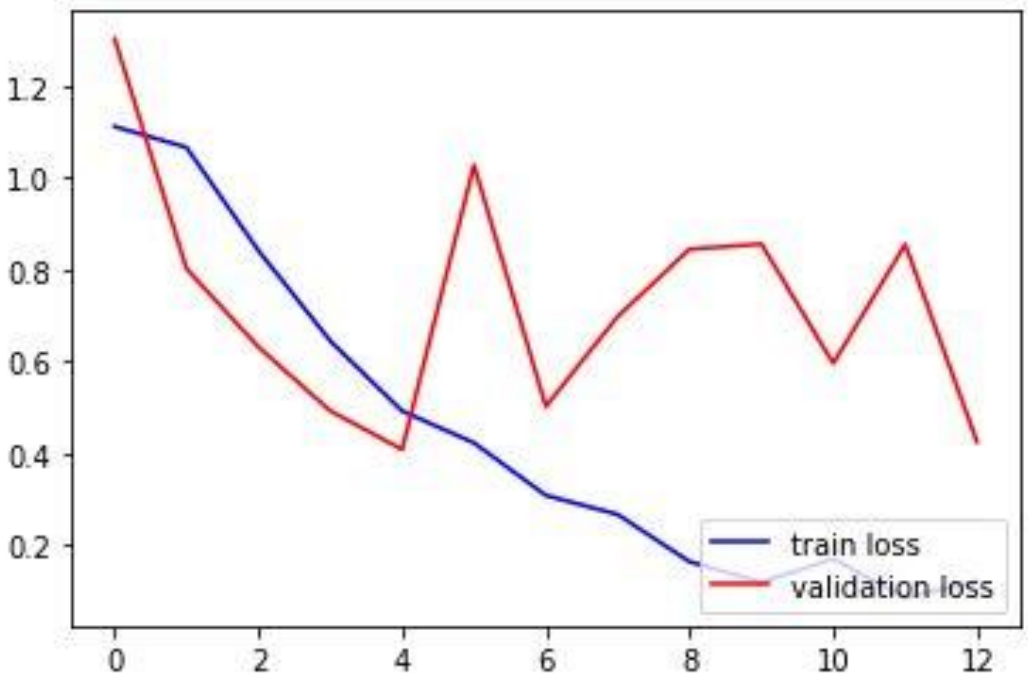

Figure 18: Training and Validation Loss for GloVe LSTM

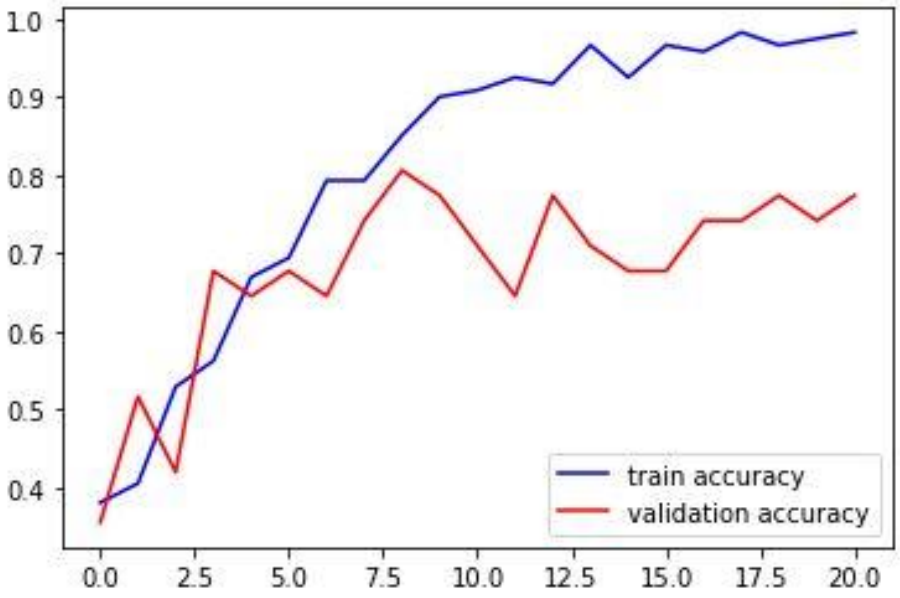

Figure 19: Training and Validation Accuracy for GloVe LSTM

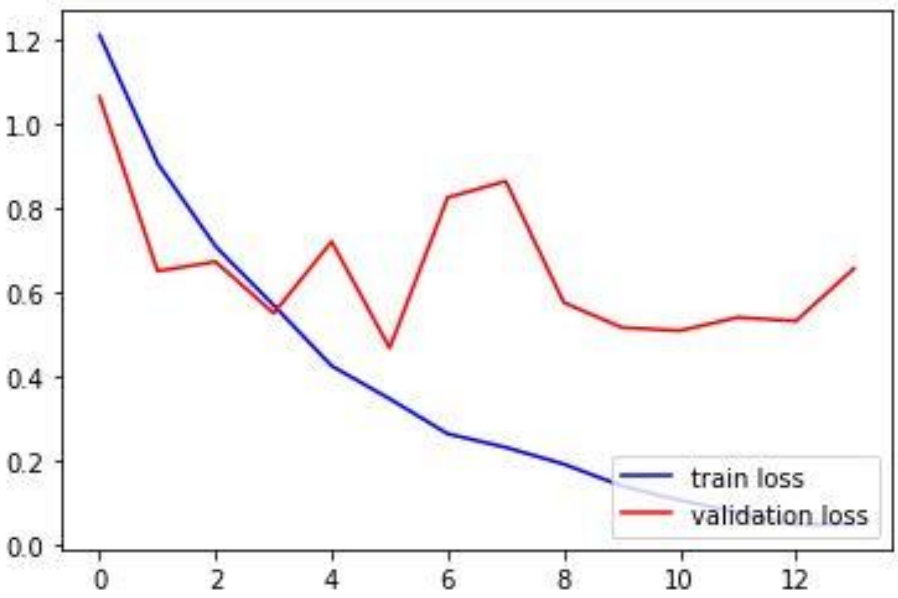

Figure 20: Training and Validation Loss for GloVe GRU

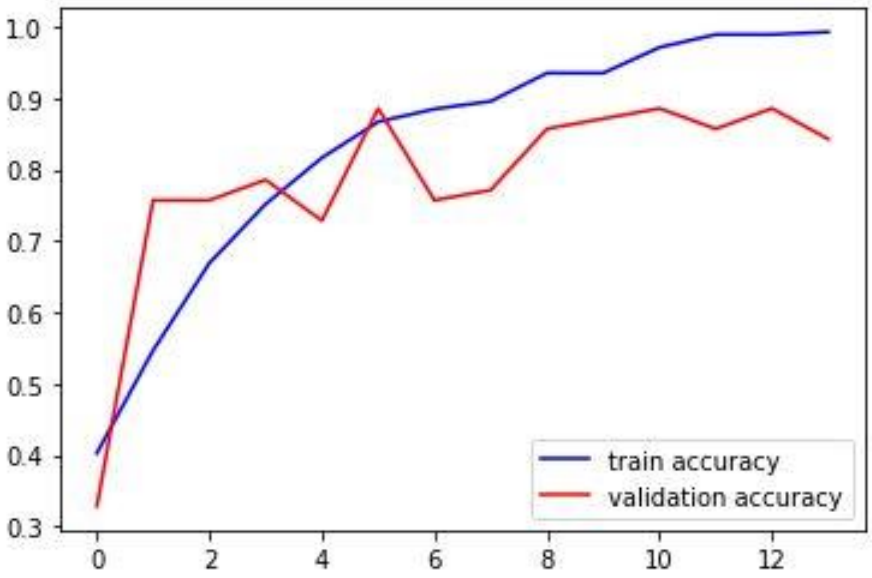

Figure 21: Training and Validation Loss for GloVe GRU

5.6 Inferences Drawn

- Different extensions to the dataset can be added to get varied functionality from the project.
- Most of the outputs of the model are brilliant, but with increase in number of charges the accuracy drops.
- To get the best functionality out of the model, we might need a huge dataset to cover all the sections of the IPC. But that requires a lot more resources and training than what is currently possible.

5.7 Validation of Objectives

Table 9: Validation of Objectives

S.No	Objectives	Status
1	To study the judicial system of India and the Indian Penal Code along with its sections in detail.	**Yes**
2	To create a Machine-Learning model that can be used for Automatic Charge Identification based on the facts of a situation, and for the prediction of relevant IPC sections based on these identified charges.	**Yes**
3	To develop an online platform for all users to access this model for their specific purposes.	**Yes**
4	To test the Model on real-world legal cases and test the accuracy for the same.	**Yes**

6. CONCLUSIONS AND FUTURE DIRECTION

6.1 Conclusions

With this project's machine learning and deep learning principles applied to real-life datasets, quantified results are obtained. We proposed a model which uses RNN, LSTM and GRU for Charge identification and Name Entity Recognition for IPC section matching. Our System includes an input for case facts and predicts the legal charges based on those facts, and furthermore matches these legal charges to the relevant IPC sections as output.

6.2 Environmental, Economic and Societal Benefits

Machine Learning in Law is a relatively new concept and this project has the potential to make a difference in the legal world.

- Justice is delayed as a result of the constant overload of cases that legal teams and prosecutors must handle. This tool will be useful for those who are short on time and resources, as well as serving as a support for law firms.
- Law students can also use this application, giving them more practical experience while still in school. These encounters would benefit their mental development and aid with information retention.
- Police can also use this, which will prevent more arbitrary arrests.
- The tool will be easy to use, making it practical for use by the general public as well.

6.3 Reflections

This project started with the direction in mind to integrate machine learning into the legal world and to create a solution for delayed justice, and to build a tool that would be informational for all citizens at a grassroot level and act as an assistant for law firms and professionals.

- Case Files contain a lot of noise and require legal knowledge for understanding. Thus, it takes time to work through these case files before building any application around it.
- This project helped us understand the legal system of India and gave us an overview of the Indian Penal Code.
- This project also helped us understand the importance of software engineering principles like testing, design specifications and effective documentation.

- While building the project, the whole team realised the importance of properly commented code and version control. We were also able to devise internal protocols for cohesive software development.
- Since the project involved research work, we learnt about research methodologies and the process of writing research articles.

6.4 Future Work

- Including more IPC sections in the model for inclusion of a larger variety of cases.
- Extension of website to support different views for different stakeholders.
- Collaborating with various law firms to include features according to their use cases.
- Collaborating with law universities to include features according to the requirement of their students.
- Exhaustive testing and professional deployment of the project website.
- Incorporating diverse data and improving model generalization.

7. PROJECT METRICS

7.1 Challenges Faced

- Communication between team members: Due to the placement season and the anxiety that comes with it, the team was not able to gather at one location for the most part of the project duration. Online meetings helped in the process, still, it was observed that remote meetings were less efficient than the ones together on campus.

- System Compatibility Issues: All team members had different operating systems in their respective laptops. This caused compatibility issues for some libraries and software due to development work.

- Product Development Technique: The website development work had a tight deadline. Agile methodology was found to be helpful for rapid product development.

- Lack of legal knowledge: As computer science students, it was a struggle to understand case files and legal processes. This was overcome by the help of the legal department of the university as well as lawyers that we were in touch with through personal contacts.

7.2 Relevant Subjects

Table 10: Relevant Subjects

Subject Code	Subject Name	Description
UCS407	Invention and Innovation	Opened new horizons for creative thinking and problem solving. Inspired to participate in Innovation.
UCS503	Software Engineering	Provided fundamental knowledge of agile methodology for efficient software development.

UML501	Machine Learning	Helped learn the basics of efficient Machine Learning
UCS633	Data Analytics and Visualisation	Helped produce insights from data.
UML602	Natural Language Processing	Primary focus of the project involved using natural language text to obtain useful information.
UCS742	Deep Learning	Helped learn the basics of efficient Deep Learning Models.
UCS310	Database Management Systems	Knowledge of databases was essential for building an efficiently working end product.
UCS408	Statistics and Probability	Provided a solid foundation for Machine Learning and Data Analytics.

7.3 Interdisciplinary Knowledge Sharing

In this project, we have used the principles of database managing to build a database. For the low level implementation, we used MySQL, which is an open-source and one the most trusted database management system. For front end, the implementation we used JavaScript. As the choice of the scripting language, used to connect our front end and back end, we used Django.

7.4 Peer Assessment Matrix

Table 11: Peer Assessment Matrix

		Evaluation of			
		Ayush	**Anika**	**Abhinav**	**Taranum**
Evaluated By	**Ayush**	5	5	5	5
	Anika	5	5	5	5
	Abhinav	5	5	5	5
	Taranum	5	5	5	5

7.5 Role Playing and Work Schedule

- **Ayush Gupta**: Model Training, Dataset Collection and Pre-Processing
- **Anika Shukla**: Machine Learning, Research, Documentation
- **Abhinav Gupta**: UI development, Dataset Collection, Research, Deployment
- **Taranum Preet Kaur**: Dataset Collection and Pre-Processing, Research

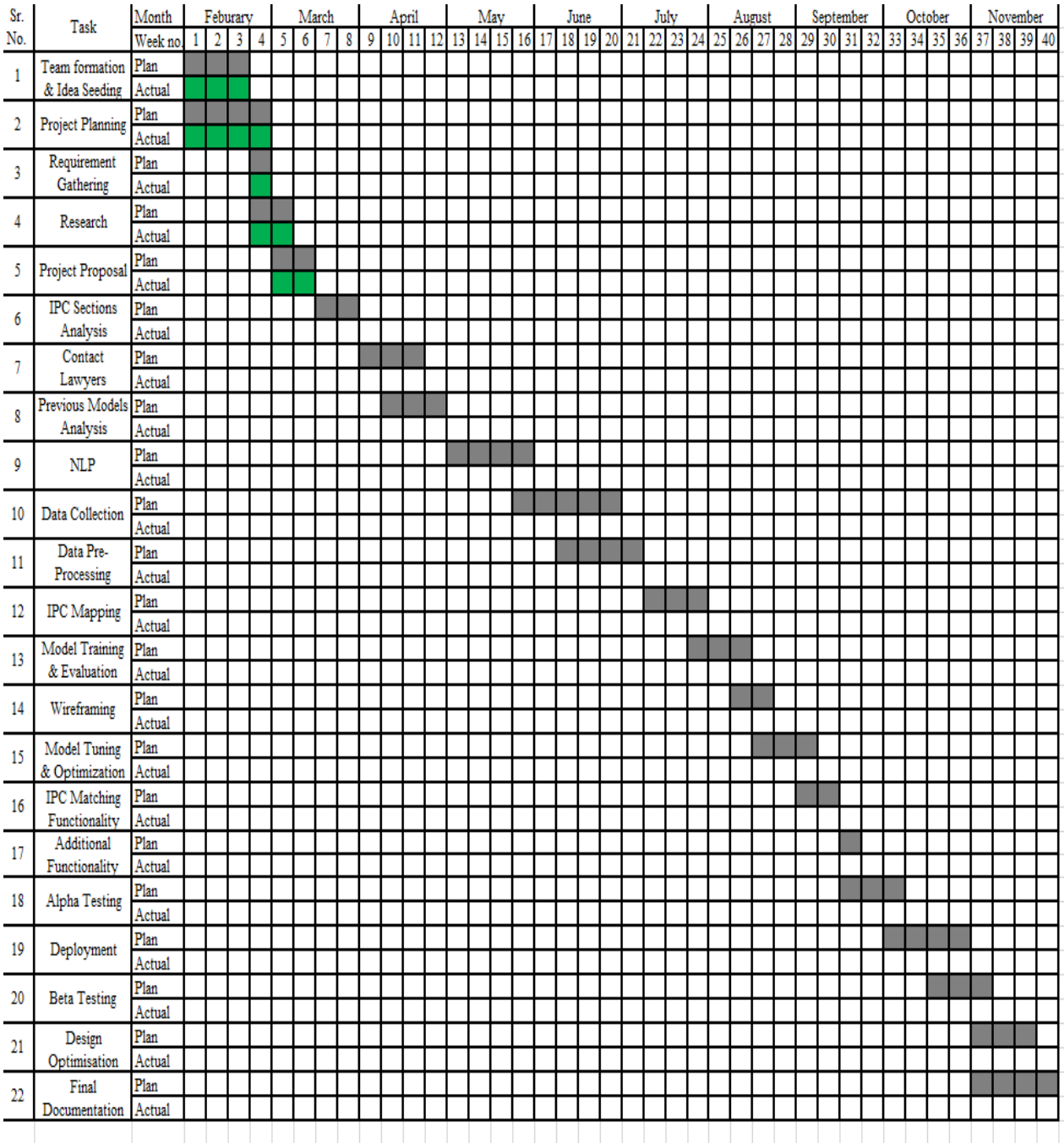

Figure 14: Work Schedule

7.6 Student Outcomes Description and Performance Indicators (A-K Mapping)

Table 12: SO and Performance Indicators

	Description	Outcome
AI	Applying mathematical concepts to obtain analytical and numerical solutions	The proposed model, BERT is a deep learning model in which every output element is connected to every input element, and the weightings between them are dynamically calculated based upon their connection.
BI	Identify the constraints, assumptions and models for the problems.	• It is assumed that all the facts are being put into the system are accurate for an accurate prediction.
B2	Use appropriate methods, tools and techniques for data collection.	We used the supreme court website for the collection of our original data. Data was manually annotated by reading and understanding the context of each case file carefully.
B3	Analyse and interpret results with respect to assumptions, constraints and theory.	• Model was able to identify legal charges with relevant IPC sections context. Tweets which used obscure words were misclassified. • Since the dataset was small, overfitting was observed on words with very less frequency.

C1	Design software system to address desired needs in different problem domains.	Our system was made to streamline collaboration between various entities with special attention towards classification accuracy.
D1	Fulfil assigned responsibility in multidisciplinary teams.	The whole project was divided among four team members to make effective use of the shared knowledge.
D2	Can play different roles as a team player.	Each member was easily able to switch from one responsibility to another responsibility to the need. Documentation and development responsibilities were shared among all the members.
E1	Identify engineering problems.	As websites grow older, we would have to improve our storage solutions to store accumulated data from various cases.
F1	Showcase professional responsibility while interacting with peers and professional communities.	All of the members were punctual in tending the group meetings to discuss the upcoming responsibilities. The team was regular at arriving at the evaluation destination for panel and mentor evaluation as well.
G1	Produce a variety of documents such as laboratory or project reports using appropriate formats	The team has been successful in preparing and presenting the project documentation in appropriate format.

G2	Deliver well-organized and effective oral presentation.	The team has been able to effectively communicate the idea behind the project and its implementation details to the mentor and the evaluation panel.
I1	Able to explore and utilize resources to enhance self learning.	The team relied on different research papers and websites like medium, Quora, Youtube to help with the implementation and theoretical concepts of the project. MOOCs were also consulted to help set up the basics.
K1	Write code in different programming languages.	Various programming languages were used for development of the project such as Python, JavaScript, HTML, CSS.
K2	Apply different data structures and algorithmic techniques.	Deep learning frameworks follow a traditional concept of forming a graph using pre-created modules. Hence, the whole architecture was formed taking into account the working of graphs.
K3	Use software tools necessary for computer engineering domain	Tensorflow and Keras, frameworks were used in Python 3.0, while using CUDA with an Nvidia GPU on Windows machines. Flask and Bootstrap were used for website development.

7.7Brief Analytical Assessment

Machine Learning in Law is interesting and useful, but current ML-AI systems are still far from this goal. However, in recent years generic and powerful neural network architectures have been developed to process natural language. The model training requires a lot of time and patience due to heavy datasets in machine learning for an extensive period of time. Furthermore, the legal system of India is quite in-depth and requires a lot of knowledge before incorporating machine learning into it. With over 512 sections, it leads to a vast dataset with massive training.

We developed a simple and effective model that automatically identifies legal charges based on the facts of a situation and matches them to the relevant IPC sections. The IPC sections targeted for this project were 302, 366 and 376. The model is trained on this dataset, and we demonstrate its performance both on the training set categories and the testing set. Hence we demonstrated that the model can predict relevant IPC sections accurately.

- **Sources of Information:** The team started reading a number of research papers and blogs to explore the current problems which could be turned into the Capstone project. Talking to people around us as well gave us a number of ideas. Some of the ideas were discarded when their actual implementations were beyond our expertise or when they didn't fulfil the requirements of the capstone project. After many deliberations, Automatic Charge Identification and IPC section matching was chosen as our capstone.

- **Analytical, computation and/or experimental methods used**: Proper working of our project and correct section generation for particular text was validated using validate dataset and refinement is done by tuning hyper-parameters.

- **Knowledge of engineering principles:** The successful completion of the project required basic knowledge of Deep learning concepts, Natural Language processing,

Multi Neural Perceptron. In addition, we also learnt about NLTK, Lime, Scikit learn, TensorFlow and Keras for our project.

- **Sharing of responsibilities**: To coordinate design and manufacturing dependencies, we divided the responsibilities among the four of us and used to meet at least once a week to update each other about the progress. Most of the work of the project was done at meetings. The documentation was shared through Google Docs among all the members for consistency.

- **Appreciation for problem-solving:** Working on the capstone project gave us a simulation of the actual problem-solving practices. We came to learn about the importance of proper documentation and requirement analysis. We also came to understand the importance of dividing up the task into different independent modules and linking them together at the end. This project also gave us deep insight into our subjects.

APPENDIX 1: REFERENCES

[1] Shirsat, K., Keni, A., Chavan, P., & Gosavi, M. (2021). Legal Judgement Prediction System.

[2] Shelar, A., & Moharir, M. (2021). Predicting Outcomes of Court Judgments-A Machine Learning Approach. In 2021 International Conference on Intelligent Technologies (CONIT) (pp. 1-6). IEEE.

[3] X. Li, X. Kang, C. Wang, L. Dong, H. Yao and S. Li, "A Neural-Network-Based Model of Charge Prediction via the Judicial Interpretation of Crimes," in IEEE Access, vol. 8, pp. 101569-101579, 2020, doi: 10.1109/ACCESS.2020.2998108.

[4] Shounak Paul, Pawan Goyal, and Saptarshi Ghosh. (2020). Automatic Charge Identification from Facts: A Few Sentence-Level Charge Annotations is All You Need. In Proceedings of the 28th International Conference on Computational Linguistics, pages 1011–1022, Barcelona, Spain (Online). International Committee on Computational Linguistics.

[5] Pillai, V. G., & Chandran, L. R. (2020). Verdict Prediction for Indian Courts Using Bag of Words and Convolutional Neural Network. In 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT) (pp. 676-683). IEEE.

[6] Ye, H., Jiang, X., Luo, Z., & Chao, W. (2018). Interpretable charge predictions for criminal cases: Learning to generate court views from fact descriptions. arXiv preprint arXiv:1802.08504.

[7] Hu, Z., Li, X., Tu, C., Liu, Z., & Sun, M. (2018).Few-shot charge prediction with discriminative legal attributes. In Proceedings of the 27th International Conference on Computational Linguistics (pp. 487-498).

[8] Xiao, C., Zhong, H., Guo, Z., Tu, C., Liu, Z., Sun, M., ... & Xu, J. (2018). Cail2018: A large-scale legal dataset for judgment prediction. arXiv preprint arXiv:1807.02478.

[9] Sinha, K., Dong, Y., Cheung, J. C. K., & Ruths, D. (2018). A hierarchical neural attention-based text classifier. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (pp. 817-823).

[10] Luo, B., Feng, Y., Xu, J., Zhang, X., & Zhao, D. (2017). Learning to predict charges for criminal cases with legal basis. arXiv preprint arXiv:1707.09168.