

Bio-Code

By Arabinda Chand



roll no.-11

S.E.(sem-4)

Computer engineering

**MINI PROJECT**

Report on

**“Bio-Code”**

Submitted in partial fulfillment of the requirements of the degree

**BACHELOR OF ENGINEERING IN**COMPUTER ENGINEERING

By

**ARABINDA CHAND**

Guide

### Dr.Shital Agrawal



**Department of Computer Engineering**

**B R Harne College of Engineering and Technology**

**Karav, Vangani (W) -421503**

**University of Mumbai (AY 2024-25)**

**CERTIFICATE**

This is to certify that the Mini Project entitled **“Bio-Code”** is a bonafide work of carried out by his/her under the supervision of Prof. Shital agrawal, and it is submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **“Bachelor of Engineering”** in **“Computer Engineering "..**

**(Prof. Shital Agrawal)**

Guide

**( Dr. Shital Agrawal ) ( Dr. Vikram Patil )**

Head of Department Principal

**Mini Project Approval**

This Mini Project entitled “**Bio-Code”** by **Arabinda Chand** is approved for the degree of **Bachelor of Engineering** in Computer Engineering **.**

**Examiners**

**1………………………………………**

(Internal Examiner Name & Sign)

**2…………………………………………**

(External Examiner name & Sign)

Date:

Place: KARAV

**Contents**

**Abstract ........................................................................................................................ 1**

**Acknowledgement ........................................................................................................ 2**

**List of abbreviations .................................................................................................... 3**

1. **Introduction ...................................................................................................... 4** 
   1. Introduction **4**
   2. Motivation **5**
   3. Objectives **5**
2. **Literature Survey ............................................................................................ 6-7** 
   1. Survey of Existing System **6**
   2. Limitation Existing system or research gap **6**
   3. Mini Project Contribution **6**
3. **Problem definition .......................................................................................... 7**
   1. Problem Definition **7**
4. **Proposed System .......................................................................................... 8-15**
   1. Architecture framework 8-**9**
   2. Algorithm **9-15**
5. **Planning And Time Estimation ................................................................. 16**
6. **System Requirement…................................................................................ 17**
   1. Details of Hardware  **6.2**Details of Software **17**

**System Design .................................................................................... 18**

* 1. Data Flow Diagram **18**

**6.4** Shared Preferences **18**

1. **Implementation ...................................................................................... 19**
2. **Testing…................................................................................................. 20-21**
   1. System Performance **20**
   2. Observation **21**
   3. Test Case **21**
3. **Result Analysis… ....................................................................................... 22-23**
4. **Conclusion And Future Scope .................................................................. 24**

# ACKNOWLEDGEMENT

There are times when words cannot express our gratitude. It is with us, as we cannot find appropriate words that would express our deep sense of gratitude and satisfaction. We would like to take this opportunity to thank everybody who helped through the successful completion of this project. Many people have contributed to our achievements during the project and we take this opportunity to thank each one of them at end of the project durations.

First, we would like to thank our Principal Sir, H.O.D & our project guide Prof.Shital Agrawal, who has extended all his valuable guidance, help and constant encouragement through the various difficult stages involved in the development of the project.

We would like to express our heartfelt thanks to all who contributed their valuable experience and knowledge and provided constant guidance to us. Thus, we feel fully obliged and convey our thanks to the teaching as well as non-teaching staff of the department. Last but not the least we would like to thank to our project partners(books) for their efforts and help.

Last but not the least we are thankful to our parent and friends for their constant Inspiration, encouragement and well wishes by which we have made a challenging project.

**Terms**

IPR-intellectual property rights are rights related to restriction of commercial use of literary works,inventions and discoveries without the permission of author

Biopatent-A right for inventors by government to prevent others from commercial use of their biological invention

Copyright(©)-using copyrighted work without consent infringing intellectual property without credits

Trademark infringes when a mark used is identical or deceptively same or confusing to registered ones

Plagiarism’s stealing others thought or language by unauthorized imitative use representing as own:wrong citation,no credits,incorrect use,lifting & modifying & no collab credits,to avoid this offence-give credit to quote,paraphase & ideas

HTTPS-hypertext transfer protocol secured-rules of internet to transfer multimedia

Web browser-A software used to connect & work with internet accessing & connecting webpages,sending & receiving information through web server

Website is collection of related web documents to common theme serving webpages

World wide web-collection of information stored on computers around the world

Webpage-files on websites containing text,graphic & sound connected by hyperlink(text jumping webpage)

URL-uniform resource locator-unique address of webpage on internet

Domain-type of organisation’s website

.com-commercial organisations, .gov-government organisation, .org-non-profit organisation, .net-networking organisation, .edu-educational organisation

RSS-really simple syndication is a program read by computer to suggest content as per subscribed websites

Hypertext Markup Language-a standard language used to build web pages, organizing and structuring content through special codes known as markup, tags that define the different parts of a webpage

Web apps-application run directly on cloud without need to install

Object oriented program(OOP) prefers objects built around & accessed by functions with bottom up approach & filled with data with similar structure in category

JVM(Java virtual machine) is an interpreter & main component of java translate byte to machine code

JDK(Java development kit) is compiler toolkit to compile java program with few classes in packages

Database is organized collection of interrelated data serving applications & database management system is general purpose software defining,constructing & manipulate them & when organized as tables collected n related by values becomes relational database management system where structured query language interface create,access/retrieve & manipulate/transform databases

**CHAPTER NO.01**

# INTRODUCTION

## 1.1 Introduction:

**In business these are the principles to come up with ideas Research & Development-I have referred several books for this project &projects before,intrapreneuring-I feel I belong to bioinformatics,hobbies or interests-this is the field im passionate about,need recognition-bioinformaticians have different needs,improve existing product-refer literature survey,IDEAS=Identifying opportunities,Designing prototype,Exclusive or unique,Acceptable to society & Satisfying to target segment.its an patenting website that simplifies your patenting journey,achieving what every engineer wants to achieve,to automate process,so that humans work on higher developmental goals and I also want young minds start thinking like scientists from a young age too so that,they grow into greater scientists.as patenting lets you own your invention incentivizes & motivates you to achieve greater intellect,So, I want to make barrier free like social media for talent & shopping accessible on handset by ecommerce,patenting is time,money &effort draining so, it needs efficiency,I vision it reaching all domain of science & art**

**1.1.2 Benefit:**

* Patenting promotes research as it rewards researchers,lettin’em own the work as per rights
* Management shift provides peace of mind for creativity
* Patent democratization will provide many perspective,statistically high chance of quality patents &prepare young imaginative minds for this competitive spaced
* Useful for eradication of duplication of effort
* Used for tracking collecting & sharing knowledge but with dignity for the discoverer
* Unifying making universal patents
* Inspiration helps coming up with project
* You check your patentability with uniqueness checker before applying
* Store option lets you reserve your code for future actions
* Our team fills form for you,so,bye bye errors

## 

I love to make bioinformatics project but now I wanted to do something for bioinformaticians,giving back to the community,I love competitive exams for the reason they incentivize by giving proof of your work,which I don’t find in hackathon,I want to introduce it in hackathons making them trust worthy and automate patenting,so that we work on greater good for society.I felt patenting needs to be made easy & accessible to all reducing entry barriers so no,ideas go waste & we tap true human resource potential.especially young minds,that are highly imaginative.So that we get many perspectives & it needs to get preserved to pass on giving creator his rightful credits,for better growth we need centralized transparent platform with fairness promoting safe environment.patenting needs a professional help like our customer support & business may need marketplace for comfortable buying,my philosophy says good stuff should be easy,accessible,fun &rewarding so,that it gets encouraged & promoted.so,creative minds don’t get tired in unproductive paperwork & creatively brainstorm.

## 1.2 Motivation:

## 1.3 Objectives

* To make patenting smooth,less shady,efficient, interactive& accessible
* To make universally accessible patents
* To keep record of knowledge easy to pass on
* To incentivize rightfully giving credits fostering innovation culture by spreading patenting to foster blooming research minds
* Participation of all different part of society(especially bringing young pure imaginative minds on table also preparing them for future), wasting no talent tapping deeper human potential by exchange of perspective &collab of ideas
* To give facilities to business & common man in his every step of patenting like rights sell assist &code uniqueness checking

**CHAPTER NO.02**

**LITERATURE SURVEY**

**2.1 Survey of Existing System**

### I found European patenting website but it wasn’t search engine optimized Couldn’t find patent filing option in official indian govt website

### 2.2Mini project contribution -Ideation research,coding(frontend,backend & database),abstract & ppt also report

### 2.3Limitation of existing system-I found bioinfo is gonna trend & it’s the niche I love,I started working on it & hadn’t thought about checking it on internet but I surfed on net after feedback I found official Indian patent site but it was for showing researches done but couldn’t find option for filing & mostly here patents are done in Indian patent office

### Market is Fragmented: There are many patenting organisations unifying,centralizing & standardizing them is not a child’s play

### Tough to spread: Bioinformatics is a new field established in 2010’s for human genome project by US,UK,France,Germany,Japan & China & also its very specific niche bad for scalability but good for strong community foothold which can payoff when in future it booms,just need to market it as fun

### Load on system: It requires many connectors such as spark,JDBC,etc with three frontend languages & backend with DBMS

### Managing large databases: It is a tough job to pull off managing databases like ncbi & ebi & getting hands on their rights

### Conceptualization of many features: Thinking about facilities catering patenting journey from start to end takes time

* **Convincing hackathon:** Hackathons that take up rights & pay the winner a share of entry fee or a job convincing them to give up profits for trustability is sensible but draining
* **Ai designing & training:** Designing AI for professional legal help is tougher & fing data to feed it is also painful

**CHAPTER NO.03**

**PROBLEM DEFINITION**

**3.1 Problem Definition**

Good things like innovation needs participation of people from all aspects of life to contribute in perspectives to look at problem with ease of access & incentivization which promotes it boosting economy too,knowledge bank should be recorded to pass on this legacy



**CHAPTER NO.04**

**PROPOSED SYSTEM**

### 4.1Architecture/framework

**Architectural Layers:**

* **Presentation Layer (Frontend):** 
  + **This layer is responsible for the user interface and user interaction.**
  + **Technologies: HTML, CSS, JavaScript.**
  + **HTML (index.html):** 
    - **Provides the structure of the web page, including forms, modals, and the action selection menu.**
  + **CSS (Inline Styles):** 
    - **Provides basic styling for the web page.**
  + **JavaScript:** 
    - **Handles user interactions, sends HTTP requests to the backend, and updates the UI based on responses.**
    - **Uses fetch API for network requests and FormData for handling form data and file uploads.**
    - **Manages modal visibility.**
* **Application Layer (Backend):** 
  + **This layer handles the business logic, data processing, and communication with the database.**
  + **Technologies: Java (Spark), Gson, JDBC.**
  + **Spark Framework:** 
    - **Provides a lightweight web framework for handling HTTP requests and responses.**
    - **Defines routes for different actions (login, store, sell, certify, check, inspiration).**
  + **Java Classes (BiocodeBackend.java):** 
    - **Contains the business logic for user authentication, file storage, database interactions, and other functionalities.**
    - **Uses JDBC to connect to the MySQL database.**
    - **Uses Gson to serialize and deserialize JSON data.**
    - **Handles file uploads, and stores the files within the uploads directory.**
    - **Handles the random topic generation.**
  + **JDBC (Java Database Connectivity):** 
    - **Provides the API for connecting to and interacting with the MySQL database.**
  + **Gson Library:** 
    - **Used to create and parse JSON responses.**
* **Data Layer (Backend):** 
  + **This layer manages data persistence and retrieval.**
  + **Technologies: MySQL database, File system.**
* **Database (Data Layer):** 
  + **MySQL Database:** 
    - **Stores user data and invention data.**
    - **Uses two tables: users and inventions.**
  + **File System:** 
    - **The uploads folder, that stores all uploaded python files.**

### 4.2Algorithm

**1. User Authentication (Login/Signup):**

* **Input:** Username, Contact, Name, Address.
* **Process:**
  + **Check Existing User:**
    - Query the users table to check if the provided username exists.
    - If the username exists:
      * Query the users table to check if the provided contact matches the contact associated with the username.
      * If the contact matches, return a success response with user details (ID, username, contact).
      * If the contact doesn't match, return an authentication failure response.
  + **Create New User:**
    - If the username doesn't exist:
      * Insert a new record into the users table with the provided username, contact, name, and address.
      * Retrieve the generated user ID.
      * Return a success response with the new user details.
* **Output:** JSON response (success or failure) with user details or error message.

**2. Inspiration Generation:**

* **Input:** None.
* **Process:**
  + Define an array of research topics.
  + Generate a random integer within the bounds of the array length.
  + Select the topic at the randomly generated index.
* **Output:** JSON response containing the selected research topic.

**3. File Storage (Store/Sell/Certify):**

* **Input:** User ID, Uploaded Python file, (Price for Sell, isCertified boolean for certify)
* **Process:**
  + Generate a unique file name (e.g., using a timestamp and the original file name).
  + Save the uploaded file to the uploads/ directory on the server.
  + Connect to the MySQL database.
  + Insert a new record into the inventions table with the user ID, invention name, file path, and (price, or isCertified value).
* **Output:** JSON response (success or failure) with a message.

**4. Code Uniqueness Check (Check):**

* **Input:** Program Code.
* **Process:**
  + This section is currently a placeholder.
  + **To Implement:**
    - Option 1: Hashing:
      * Calculate a hash of the program code.
      * Query the inventions table to check if a file with the same hash exists.
    - Option 2: String Comparison:
      * Retrieve all stored program codes from the inventions table.
      * Compare the input program code to each stored code.
* **Output:** JSON response (success or failure) indicating uniqueness.

### A diagram of a diagram AI-generated content may be incorrect.Flow Diagram Of Project

**CHAPTER NO.05**

**PLANNING AND TIME ESTIMATION**

**Planning And Time Estimation:**

**Fig.5.1**

**CHAPTER NO.06**

**SYSTEM REQUIREMENT**

### 6.1 Details Of Software

* Html(notepad)
* Java
* Scala plugin
* Spark
* Maven framework
* Com,google.gson.2.10.2
* JDBC
* IntelliJ
* MySQL

### 6.2 Details Of Hardware

* Ryzen 5 Processor,Windows 11
* 4 GB,64-bit,RAM
* 930 GB. HDD



[This Photo](https://gouforit.com/los-3-mejores-ordenadores-portatiles-baratos-para-estudiantes/) by Unknown Author is licensed under [CC BY-SA](https://creativecommons.org/licenses/by-sa/3.0/)

**CHAPTER NO.07**

**SYSTEM DESIGN**

### 7.1 Data Flow Diagram (DFD)

### 7.2 Shared preferences

1. **Start:** The process begins.
2. **User Action:** Checks for user interaction.
3. **Login/Signup Algorithm:**
   * Gets user input.
   * Checks if the username exists.
   * If yes, checks if the contact matches.
   * Sends success or failure responses.
   * If no, inserts new user data.
   * Displays the relevant response.
4. **Inspiration Algorithm:**
   * Defines a topic array.
   * Generates a random index.
   * Retrieves a topic.
   * Sends it as JSON.
   * Displays the topic.
5. **File Storage Algorithm (Store/Sell/Certify):**
   * Gets user ID and file.
   * Generates a unique file name.
   * Saves the file to the server.
   * Inserts invention data into the database.
   * Sends success/failure messages.
   * Displays the response.
6. **Code Uniqueness Check Algorithm:**
   * Gets the program code.
   * Calculates the hash or performs string comparison.
   * Checks the database for the hash/code.
   * Sends success/failure messages.
   * Displays the result.
7. **Backend, Database, and Frontend Subgraphs:**
   * Clearly separates backend processing, database interactions, and frontend displays.
8. **Looping:**
   * The flowchart loops back to "User Action?" to handle subsequent user interactions.

This flowchart visually represents the algorithms used in your "Biocode" project, providing a clear understanding of the process flow for each functionality.

**CHAPTER NO.08**

### IMPLEMENTATION

 **User Credentials:** Frontend -> Backend -> Database (users table).

 **Uploaded Files:** Frontend -> Backend -> File System -> Database(inventions table).

 **Invention Data:** Frontend -> Backend -> Database (inventions table).

 **Responses:** Backend -> Frontend (JSON).

A screen shot of a computer program

AI-generated content may be incorrect.A screen shot of a computer

AI-generated content may be incorrect. **Inspiration Data:** Backend -> Frontend (JSON).

A computer screen with text on it

AI-generated content may be incorrect.A computer screen shot of a computer code

AI-generated content may be incorrect.A screen shot of a computer code

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.A black background with white text

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

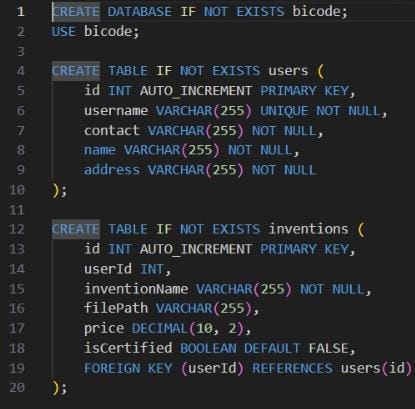
A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.A screen shot of a computer screen

AI-generated content may be incorrect.



**Chapter 09**

**TESTING**

**9.1 System performance**

1. User Authentication (Login/Signup):

* Database Query: The core operation is a database query to check if the username exists and then another to check the contact. In a well-indexed database, these queries would typically be O(log n), where n is the number of users.
* Overall: The dominant operation is the database query, so the overall Big O notation for user authentication is O(log n).

2. Inspiration Generation:

* Array Access: This involves retrieving a random element from an array. Array access is a constant-time operation.
* Overall: The Big O notation for inspiration generation is O(1).

3. File Storage (Store/Sell/Certify):

* File I/O: Saving the file to the file system is dependent on the file size. Let's assume the file size is 'm'. This would be O(m).
* Database Insertion: Inserting a record into the database is typically O(log n) for a well-indexed table.
* Overall: The dominant operation is the file I/O, so the overall Big O notation is O(m), where 'm' is the size of the uploaded file.

4. Code Uniqueness Check:

* Hashing Method:
  + Calculating the hash of the program code is O(k), where k is the size of the code.
  + The database query to check for the hash is O(log n).
  + Overall: The Big O notation would be O(k + log n). If the code size is relatively small compared to the number of database entries, O(log n) could dominate. If the code size is large, then O(k) would dominate.
* String Comparison Method:
  + Retrieving all program codes from the database is O(n), where n is the number of inventions in the database.
  + Comparing the input code to each stored code is O(k \* n), where k is the size of the input code.
  + Overall: The Big O notation would be O(k \* n). This is significantly less efficient than the hashing method.

Summary:

* User Authentication: O(log n)
* Inspiration Generation: O(1)
* File Storage: O(m) (where 'm' is file size)
* Code Uniqueness Check:
  + Hashing: O(k + log n) (where 'k' is code size, 'n' is number of inventions)
  + String Comparison: O(k \* n) (where 'k' is code size, 'n' is number of inventions)

**9.2 Test case**

1. User Authentication (Login/Signup):

* Test Case 1: Successful Signup
  + Steps:
    1. Navigate to the signup form.
    2. Enter valid username, contact, name, and address.
    3. Submit the form.
  + Expected Result:
    1. User is successfully registered.
    2. User is logged in.
    3. Success message is displayed.
    4. User data is stored in the database.
* Test Case 2: Successful Login
  + Steps:
    1. Navigate to the login form.
    2. Enter valid username and contact.
    3. Submit the form.
  + Expected Result:
    1. User is successfully logged in.
    2. Success message is displayed.
* Test Case 3: Invalid Login Credentials
  + Steps:
    1. Navigate to the login form.
    2. Enter invalid username or contact.
    3. Submit the form.
  + Expected Result:
    1. Login fails.
    2. Error message is displayed.
* Test Case 4: Username Already Exists (Signup)
  + Steps:
    1. Navigate to the signup form.
    2. Enter a username that already exists.
    3. Submit the form.
  + Expected Result:
    1. Signup fails.
    2. Error message indicating username already exists.
* Test Case 5: Empty Fields (Signup/Login)
  + Steps:
    1. Navigate to the signup/login form.
    2. Leave some fields empty.
    3. Submit the form.
  + Expected Result:
    1. Form submission is prevented (client-side validation).
    2. If not, server-side validation should return an error.

2. Inspiration Generation:

* Test Case 6: Get Inspiration
  + Steps:
    1. Click the "Get Inspiration" button.
  + Expected Result:
    1. A random research topic is displayed in an alert or on the page.

3. File Storage (Store/Sell/Certify):

* Test Case 7: Successful File Storage (Store)
  + Steps:
    1. Select "Store Invention" action.
    2. Upload a valid Python file (.py).
    3. Submit the form.
  + Expected Result:
    1. File is successfully stored on the server.
    2. Invention data is stored in the database.
    3. Success message is displayed.
* Test Case 8: Successful File Storage (Sell)
  + Steps:
    1. Select "Sell Invention" action.
    2. Upload a valid Python file (.py).
    3. Enter a valid price.
    4. Submit the form.
  + Expected Result:
    1. File is successfully stored on the server.
    2. Invention data, including price, is stored in the database.
    3. Success message is displayed.
* Test Case 9: Successful File Storage (Certify)
  + Steps:
    1. Select "Certify & Patent" action.
    2. Upload a valid Python file (.py).
    3. Submit the form.
  + Expected Result:
    1. File is successfully stored on the server.
    2. Invention data, including the certification flag, is stored in the database.
    3. Success message is displayed.
* Test Case 10: Invalid File Type
  + Steps:
    1. Select any file storage action.
    2. Upload a file with an invalid extension (e.g., .txt, .jpg).
    3. Submit the form.
  + Expected Result:
    1. Client-side validation prevents file upload.
    2. If not, server-side validation should return an error.
* Test Case 11: File Upload Size Limit
  + Steps:
    1. Select any file storage action.
    2. Upload a file exceeding the allowed size.
    3. Submit the form.
  + Expected Result:
    1. Error message indicating file size limit exceeded.

4. Code Uniqueness Check:

* Test Case 12: Unique Code
  + Steps:
    1. Select "Check Uniqueness" action.
    2. Enter a unique code snippet.
    3. Submit the form.
  + Expected Result:
    1. Success message indicating code is unique.
* Test Case 13: Non-Unique Code (Hash Method)
  + Steps:
    1. Select "Check Uniqueness" action.
    2. Enter a code snippet that already exists in the database (or has a matching hash).
    3. Submit the form.
  + Expected Result:
    1. Error message indicating code is not unique.
* Test Case 14: Non-Unique Code (String Comparison)
  + Steps:
    1. Select "Check Uniqueness" action.
    2. Enter a code snippet that is identical to one already in the database.
    3. Submit the form.
  + Expected Result:
    1. Error message indicating code is not unique.
* Test Case 15: Empty Code Input
  + Steps:
    1. Select "Check Uniqueness" action.
    2. Submit the form without entering any code.
  + Expected Result:
    1. Error message indicating code is required.

5. General Tests:

* Test Case 16: Database Connection Errors
  + Steps:
    1. Simulate a database connection error.
    2. Perform any action that requires database access.
  + Expected Result:
    1. User-friendly error message is displayed.
    2. Application handles the error gracefully.
* Test Case 17: File System Errors
  + Steps:
    1. Simulate a file system error (e.g., write permission issues).
    2. Perform any file upload action.
  + Expected Result:
    1. User-friendly error message is displayed.
    2. Application handles the error gracefully.
* Test Case 18: Cross-Browser Compatibility:
  + Steps:
    1. Test the application in different browsers (Chrome, Firefox, Safari, Edge).
  + Expected Result:
    1. Application functions correctly in all browsers.
* Test Case 19: Responsive Design:
  + Steps:
    1. Test the application on different screen sizes and devices.
  + Expected Result:
    1. The user interface adapts to different screen sizes.

**A close up of a code

AI-generated content may be incorrect.9.3 Observations**

**CHAPTER NO.10**

**RESULT ANALYSIS**

**A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.**

**CHAPTER NO.11**

**CONCLUSION AND FUTURE SCOPE**

**Conclusion: It should spread patenting to every person with imaginative mind & let the exchange of information with right credits incentivizing & raising economy flawless with input of different perspectives with making it effortless to manage their patent library,reducing their stage,providing professional accurate help,making it social media or ecommerce platform of patenting so that we don’t lose another basmati rice patent**

**Future Scope :We can connect ebi & ncbi to our uniqueness checking database.we can collab with Government to make their system smoother,We can charge recurrent fee for new patents or introduce subscription model for many patents taken care by our customer support who asks you every detail on call,without offline contact effort,you just ideate,leaving stress part to us.also we may provide you dashboard to manage all your patents. making it both D2C & B2B businesswe can collab with hackathon organizations to make them trustworthy,I want to reduce entry barrier of patent like social media so,that all can document their work & knowledge shared & want to make patenting accessible like ecommerce make shopping on handset which promotes more productive stuff reducing friction,we can promote these in schools as competitions,we may expand to other domains of science by accepting other file types,We may introduce AI in inspiration tab to tell you current trending researches,or for businesses choosing right rights to buy from huge library of rights planned in future & form simplification with future updates on website for free**

**REFERENCES**

**Book:**

Cambridge social sciences by Devyani kapoor

Trueman’s biology by K.N. Bhatia

Click start computer computer science Cambridge by n meera

Exploring information technology macmilan by rohit khurana

Dbms using sql by arahant neetu gaikwad

OOP java by techknowledge

Boot-click enter macmilan by Gurpreet bindra

Fast forward oxford by sangeeta Panchal

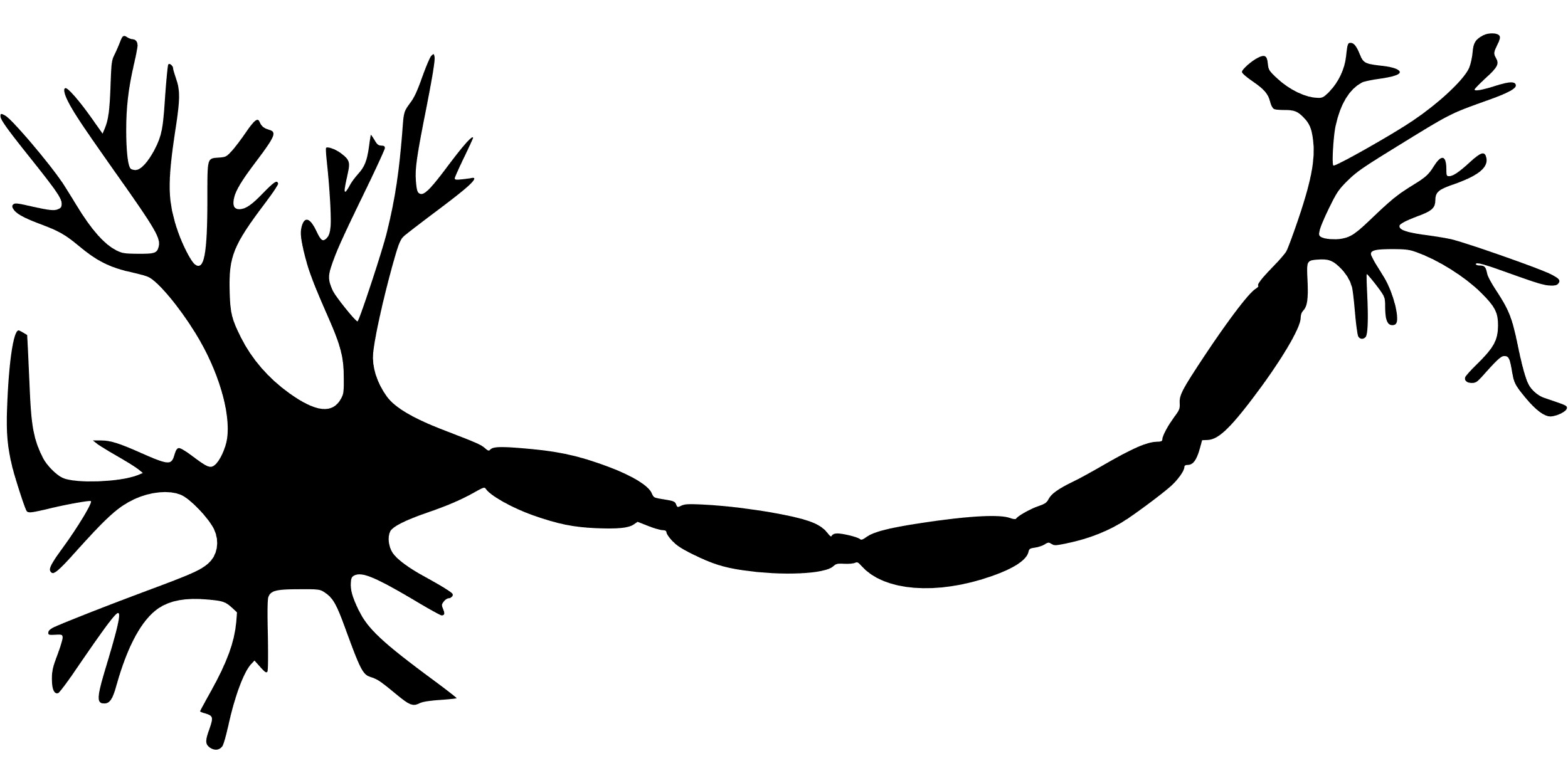
Terabytes connect with computers Cambridge by Vaishali sharma

KIPS information technology vocational

BEGS-186 Business communication by dr. anand Prakash

Bcos-185 entrepreneurship prof. Madhu tyagi

Computer science with python Preeti & sumita arora



**Thank you!!**