AIRCANVAS-DRAW WITH HANDS GESTURES

MINI PROJECT

Bachelor of Computer Applications (BCA AI-DS SEC:B)

Under the Supervision of:

Submitted by

KHUSHI 2401201102

TEESHA 2401201039

DEEPAK 2401201071

VIVAN 2401201099

ACKNOWLEDGEMENT

We, the project team, would like to sincerely thank our guide MOHAMMAD AIJAZ for their invaluable support, encouragement, and expert guidance throughout the development of our project, “AirCanvas – Draw with Hand Gestures.” Their mentorship was instrumental in helping us stay on track and overcome challenges. We also extend our gratitude to the faculty members and the Department of Computer Science, KRMU, for providing the necessary resources and a supportive academic environment .A special thanks to our fellow group members for their teamwork, dedication, and collaborative spirit. Each member’s contribution played a key role in the successful completion of this project. Lastly, we are grateful to our friends and families for their patience, motivation, and moral support throughout this journey.

THANKYOU

DECLARATION

We hereby declare that the mini project report titled “AirCanvas – Draw with Hand Gestures” submitted in partial fulfilment of the requirements for the award of the degree of Bachelor of Computer Applications (BCA)is a bonafide record of original work carried out by us under the guidance of Mohammad Aijaz Department of Computer Science , KRMU .This project has not been submitted previously by us or any other person for the award of any degree, diploma, or similar title in any university or institution.

Submitted by:

Khushi 2401201102

Teesha 2401201039

Deepak 2401201071

Vivaan 2401201099

CERTIFICATE OF ORIGINALITY

This is to certify that the mini project report titled

“AirCanvas – Draw with Hand Gestures”

submitted by the following students:

- KHUSHI 2401201102
- TEESHA 2401201039
- DEEPAK 2401201071
- VIVAAN 2401201099

to the Department of Computer Applications ,KRMU

in partial fulfilment of the requirements for the award of the degree of Bachelor of Computer Applications (BCA),is an original work carried out by them under the supervision of Mohammad Aijaz.

INTRODUCTION

In today's rapidly evolving digital landscape, human-computer interaction is becoming more natural, intuitive, and accessible. Traditional input devices such as keyboards, mice, and styluses, while effective, still create a barrier between the user and the digital world. Recognizing the growing need for more seamless and contactless interaction, we present "AirCanvas – Draw with Hand Gestures," an innovative web-based application designed to revolutionize the way users create digital art. AirCanvas enables users to draw freely in the air by simply moving their hands in front of a webcam. Utilizing the power of Computer Vision and Machine Learning, specifically the MediaPipe framework for hand tracking and OpenCV for real-time video processing, the application captures the user's hand gestures and translates them into strokes on a digital canvas. Unlike traditional drawing applications that require physical tools, AirCanvas requires only a functional webcam, thus making it highly accessible and easy to use .The project aims not only to offer a fun and creative experience but also to explore future possibilities of gesture-based systems. In an era where touchless technologies are gaining prominence—particularly after the global pandemic—solutions like AirCanvas hold immense potential across various industries including education, art, design, and entertainment.

Through a user-friendly web interface developed using Flask, HTML, CSS, and JavaScript, users can log in, start drawing

instantly, choose colors, clear the canvas, and even save their artwork to a personalized gallery. The back-end database powered by PostgreSQL ensures that user data and saved drawings are securely stored for future access. The AirCanvas project showcases the beautiful synergy between technology and creativity. It stands as a demonstration of how modern software solutions can remove physical barriers, enhance user engagement, and pave the way for more natural interaction methods. By harnessing simple hand gestures, we open a world of possibilities where digital content creation becomes more accessible, interactive, and enjoyable .This project has not only deepened our understanding of computer vision, gesture recognition, and web development but has also provided valuable insights into building responsive, real-time applications that prioritize user experience. AirCanvas thus represents a meaningful step towards the future of intuitive, touchless digital interaction.

OBJECTIVE

The primary objective of the AirCanvas project is to design and develop a web-based application that enables users to create digital drawings by interpreting hand gestures captured through a webcam.The project aims to provide a contactless, intuitive, and innovative drawing experience, enhancing the natural interaction between humans and machines.

The specific objectives of this project are as follows:

- To develop a real-time hand tracking system:

Utilize MediaPipe and OpenCV to accurately detect and track hand gestures, specifically the movement of the index finger, to create strokes on a virtual canvas.

- To offer a simple, accessible digital drawing platform:

Build a web-based interface that can be accessed through any modern web browser without the need for specialized hardware or tools, thus making it universally usable.

- To implement user authentication and data management:

Incorporate a secure login and registration system allowing users to save, retrieve, and manage their artwork through a personalized account.

- To provide user-friendly drawing functionalities:

Enable users to select different colors, clear the canvas, save their creations, and view a gallery of their previously saved drawings.

- To ensure real-time responsiveness and accuracy:

Achieve minimal lag between gesture detection and canvas rendering, thereby providing a smooth and natural drawing experience.

- To enhance creativity through technology:

Leverage computer vision and machine learning techniques to push the boundaries of traditional digital art and create new opportunities for creative expression.

- To promote the concept of touchless interaction:

Address the growing demand for contactless technologies by creating a practical application that showcases the potential of gesture-based interfaces in various fields.

By fulfilling these objectives, the AirCanvas project not only demonstrates the powerful applications of Computer Vision and Web Development but also contributes to the advancement of user-centric, natural interaction technologies.

SOFTWARE AND HARDWARE USED

The development of the AirCanvas – Draw with Hand Gestures project involves a wide range of software tools and technologies, integrated to achieve real-time gesture tracking, web-based interaction, and secure data storage:

- Python 3.11+ – Core programming language used for backend development and integration with computer vision libraries.
- Flask – A lightweight Python web framework used to build the server-side of the web application.
- OpenCV (cv2) – Used for real-time video capture and frame processing from the webcam.
- Media Pipe – Google's powerful framework for real-time hand tracking and gesture recognition.
- Flask-SQL Alchemy – Object Relational Mapper (ORM) used for database interactions with Python.
- PostgreSQL – A robust and scalable relational database system used to store user data and saved drawings.

- HTML5, CSS3, JavaScript – Used to design and build a responsive and interactive front-end interface.
- Jinja2 – Templating engine integrated with Flask for rendering dynamic HTML content.
- VS Code / PyCharm – Development environments used to write and manage the source code.

Hardware Requirements

The hardware required to develop and run the AirCanvas project is minimal and cost-effective, making it accessible to a wide range of users:

- Personal Computer / Laptop – Capable of running Python and Flask server environments smoothly. Minimum specifications include:
- Processor: Intel i3 or higher
- RAM: 4 GB or higher
- Operating System: Windows / macOS / Linux
- Webcam – An essential component used to capture real-time video of the user's hand for gesture

detection. Built-in or external USB webcams are supported.

- Internet Browser – A modern browser (such as Google Chrome, Mozilla Firefox, or Microsoft Edge) to access the web application.

SYSTEM ANALYSIS

System analysis is a critical phase of project development where we understand, define, and evaluate both the existing problem and the proposed solution. It helps in identifying system requirements and planning the structure of the new system effectively.

Existing System

In traditional digital drawing applications, users typically rely on physical input devices such as:

- Mouse
- Stylus pens
- Touchscreens
- Graphic tablets

Although effective, these methods require physical contact and depend heavily on external devices. They create barriers to natural, intuitive interaction and can also be less hygienic, especially in environments requiring minimal touch interaction.Additionally, conventional systems do not leverage hand gesture

recognition or real-time contactless control, limiting accessibility and innovation.

Challenges with the Existing System

- Dependency on Physical Devices:

Users must possess or purchase specific hardware like styluses or touchscreens.

- Limited Accessibility:

People without these devices cannot participate in digital art creation easily.

- Lack of Contactless Experience:

No support for natural, touch-free interaction, especially important in post-pandemic contexts.

- Restricted Mobility:

Drawing tools restrict freehand movement, which can limit creativity.

Identification of Need

- Touchless Interaction:

A growing need for safe, hygienic, and intuitive input methods without the necessity for touch.

- Natural Gesture Recognition:

Systems that mimic human behavior through natural hand movements are more accessible and engaging.

- Low-Cost Accessibility:

Utilizing webcams and software instead of expensive stylus-based tablets.

- Innovation in User Experience:

Providing creative new ways for users to interact with digital systems.

PROPOSED SYSTEM

The proposed AirCanvas system addresses these challenges by providing a contactless drawing platform that relies on real-time hand gesture recognition using a simple webcam.

It enables users to create artwork naturally by just moving their fingers in the air.

Key features of the proposed system include:

- Web-based Drawing Application: Accessible through any modern browser.
- Real-Time Gesture Tracking: Using MediaPipe for highly accurate hand detection.
- Canvas Interaction: Map hand landmarks to drawing actions like drawing, clearing, color selection.
- User Authentication and Gallery: Users can save, view, and manage their drawings.
- Secure Data Storage: PostgreSQL database stores user information and drawing files securely.

FEASIBILITY STUDY

- Technical Feasibility:

The project uses mature, open-source technologies such as Flask, OpenCV, MediaPipe, and PostgreSQL, ensuring high technical viability.

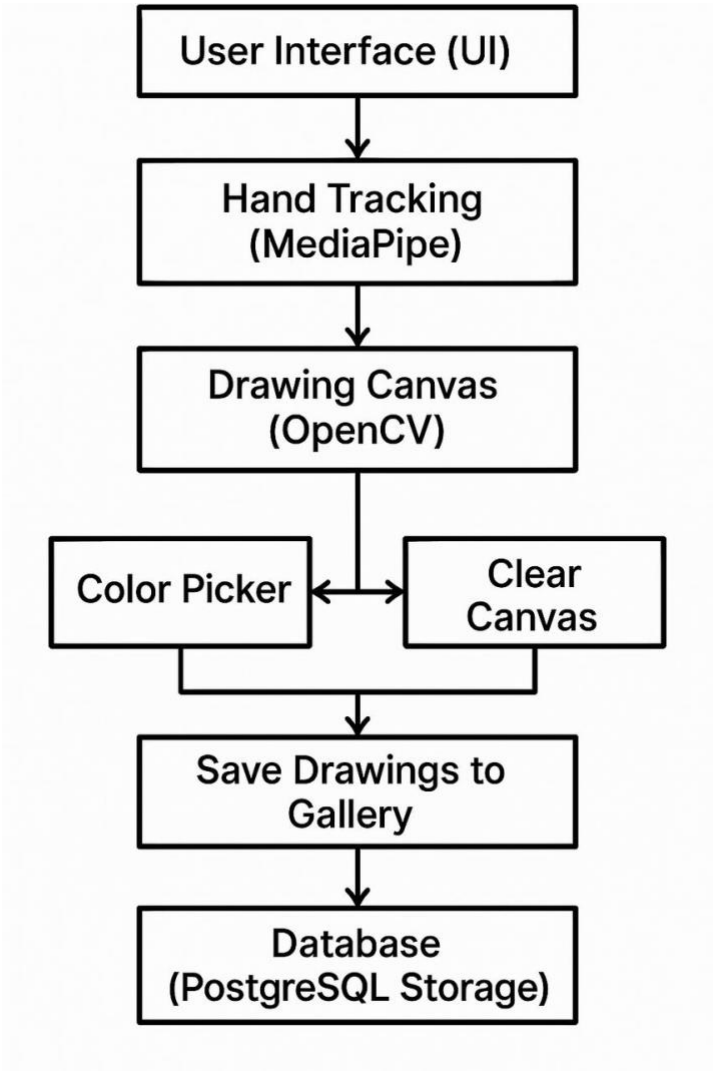
- Operational Feasibility:

The system is simple, user-friendly, and does not require complex hardware setups, making it operationally practical for a wide audience.

- Economic Feasibility:

Development costs are minimal due to the use of free, open-source libraries. Users require only basic hardware (a webcam and a computer)

MODULES



The AirCanvas system is designed using a modular architecture to ensure clarity, maintainability, scalability, and performance. Each module is responsible for a specific set of functionalities and communicates with others in a well-defined manner. The following are the major modules of the system:

1. User Interface (UI) Module

This module provides the graphical interface through which users interact with the application. Built using HTML, CSS, and JavaScript, it offers features like login/signup, drawing canvas, color selection, and gallery viewing.

Key Features:

- Web-based responsive design.
- Real-time camera preview and canvas overlay.
- Color picker and clear canvas buttons.
- User-friendly layout and minimalistic design.

2. Hand Tracking Module

This module uses MediaPipe to detect and track hand landmarks from the live webcam feed. It focuses on the position of the index finger, which is used to simulate drawing in the air.

Key Features:

- Real-time hand and finger landmark detection.
- High accuracy even with background noise.
- Determines drawing state based on gesture (e.g., raised index finger).

3. Drawing Canvas Module

Built using OpenCV, this module captures the tracked fingertip coordinates and renders the drawing on a virtual canvas. It enables users to draw dynamically in real time.

Key Features:

- Dynamic line rendering based on finger movement.
- Smooth drawing experience with minimal lag.
- Integration with color selection and clearing functions.

4. Color Picker and Canvas Controls

These tools enhance user interaction by allowing drawing customization and basic canvas management.

Key Features:

- Select drawing color from a predefined palette.
- “Clear” button to reset the canvas instantly.
- Visual indicators for active color and drawing state.

5. Save to Gallery Module

This module allows authenticated users to save their drawings. The saved files are then accessible through a personal gallery.

Key Features:

- Converts canvas drawings into image format.
- Stores drawing metadata (date, user, etc.).
- Interface to preview and manage saved artwork.

6. Authentication Module

Responsible for secure login, registration, and session management of users. Built using Flask and Flask-Login.

Key Features:

- User registration and secure login system.
- Session-based access control.
- Password hashing and security practices.

7. Database Module

This module manages persistent data storage using PostgreSQL. It stores user information, saved drawings, and other related data.

Key Features:

- Efficient and scalable relational database design.
- Data integrity and validation enforcement.
- Easy retrieval of saved drawings by user ID.

8. Export Module (Optional Enhancement)

This module enables users to export their drawings in downloadable formats like PNG or JPG.

Key Features:

- Convert current canvas to downloadable format.
- Initiate automatic download or save to local system.

CODING

```
from flask import Flask, render_template, jsonify, request, session
from flask_sqlalchemy import SQLAlchemy
from datetime import datetime
import os
from dotenv import load_dotenv

load_dotenv()

app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] = os.getenv('DATABASE_URL')
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
app.config['SECRET_KEY'] = os.getenv('SESSION_SECRET', 'dev-secret-key')

db = SQLAlchemy(app)

# Import models after db initialization
from models import User, Drawing

@app.route('/')
def home():
    if 'user_id' not in session:
        return render_template('login.html')
    return render_template('canvas.html')

@app.route('/login', methods=['POST'])
```

```
def login():
    data = request.json
    user = User.query.filter_by(username=data['username']).first()

    if user and user.check_password(data['password']):
        session['user_id'] = user.id
        return jsonify({'success': True})
    return jsonify({'success': False, 'message': 'Invalid credentials'}), 401

@app.route('/register', methods=['POST'])
def register():
    data = request.json

    if User.query.filter_by(username=data['username']).first():
        return jsonify({'success': False, 'message': 'Username already exists'}),
400

    user = User(username=data['username'])
    user.set_password(data['password'])

    db.session.add(user)
    db.session.commit()

    session['user_id'] = user.id
    return jsonify({'success': True})

@app.route('/logout')
```

```
def logout():
    session.pop('user_id', None)
    return jsonify({'success': True})

@app.route('/save-drawing', methods=['POST'])
def save_drawing():
    if 'user_id' not in session:
        return jsonify({'success': False, 'message': 'Not authenticated'}), 401

    data = request.json
    drawing = Drawing(
        user_id=session['user_id'],
        image_data=data['imageData']
    )
    db.session.add(drawing)
    db.session.commit()
    return jsonify({'success': True})

@app.route('/get-drawings')
def get_drawings():
    if 'user_id' not in session:
        return jsonify({'success': False, 'message': 'Not authenticated'}), 401

    drawings = Drawing.query.filter_by(user_id=session['user_id']).all()
    return jsonify([{
        'id': d.id,
        'imageData': d.image_data,
```

```
        'created_at': d.created_at.isoformat()
    } for d in drawings])

if __name__ == '__main__':
    with app.app_context()
  db.create_all()
    app.run(host='0.0.0.0', port=5000, debug=True)
```

OUTPUT


Welcome to Air Canvas
Login
Register
Username
Password
Login
Username
Password
Register
Air Canvas
Draw in the air using your webcam! Express your creativity with hand gestures and create beautiful digital artwork.
✨ Hand tracking technology
🎨 Multiple color options
🔄 Real-time drawing
Clear
Save
Logout
Status: Ready
Gallery
Gallery
×

TESTING

Testing Objectives:

- Verify correct gesture detection and drawing.
- Check successful login/logout and registration.
- Ensure drawings are properly saved and retrieved.

Testing Approach:

- Manual testing of gesture recognition.
- Database validation for saved drawings.
- Browser testing for UI responsiveness.

FUTURE SCOPE

- Mobile app version (Android/iOS)
- Advanced AI gesture recognition
- Real-time collaborative drawing
- Cloud storage integration
- Gesture-based shape creation (circle, square, etc.)
- Augmented Reality (AR) integration

CONCLUSION

The AirCanvas – Draw with Hand Gestures project stands as a successful integration of computer vision, gesture recognition, and web development technologies into a practical and creative application. The primary objective of providing a real-time, touchless, and intuitive drawing platform has been effectively achieved through the use of a standard webcam and advanced software libraries such as MediaPipe, OpenCV, and Flask.Throughout the development of this project, we explored the true potential of human-computer interaction by eliminating traditional input devices and replacing them with natural hand movements. The ability to draw in the air using just one's index finger demonstrates not only technical innovation but also the potential for more accessible and hygienic design, especially relevant in a post-pandemic world that values contactless technologies.

The application is capable of:

- Detecting hand gestures in real time with high accuracy,
- Translating finger movements into smooth strokes on a digital canvas,
- Allowing users to interact through an intuitive web interface,

- Saving and managing artwork using secure user authentication,
- Supporting features like color selection, canvas clearing, and gallery viewing.

The modular structure of the project has made it scalable and maintainable. Each component—from hand tracking to database management—has been developed and tested independently before integration, ensuring robustness and performance. The system was also validated through thorough testing phases, confirming its functional correctness, responsiveness, and usability.From a personal and academic perspective, the project has provided immense learning in areas such as:

- Real-time data processing and optimization,
- Handling user sessions and secure authentication,
- Integrating front-end and back-end technologies,
- Debugging real-world gesture recognition challenges,
- Designing for user experience and performance in tandem.

Moreover, this project represents the perfect example of creativity fused with technology. It not only serves a functional purpose but also encourages artistic expression in a novel, contact-free manner.

Although the current version provides all core functionalities, the potential for enhancement is vast. With additional features like undo/redo options, collaborative canvases, AI-generated shape prediction, and mobile compatibility, AirCanvas could evolve into a full-featured, cross-platform drawing tool powered by gesture recognition.

In conclusion, AirCanvas reflects innovation, practical application of emerging technologies, and a step toward more immersive and natural human-computer interactions. It has the potential to inspire future solutions in education, art, accessibility, and digital design, proving that the future of interface design lies not just in screens—but in the spaces around us.

BIBLOGRAPHY

1. MediaPipe Documentation

   https://google.github.io/mediapipe/solutions/hands

   (Used for implementing real-time hand and finger tracking features.)

2. OpenCV – Python Tutorials

   https://docs.opencv.org/

   (Used for video frame capture, drawing canvas rendering, and image processing.)

3. Flask Official Documentation

   https://flask.palletsprojects.com/

   (Used to build the web application's backend and manage routing, sessions, and templates.)

4. Flask-SQLAlchemy Documentation

   https://flask-sqlalchemy.palletsprojects.com/

   (Used for integrating Python with PostgreSQL using ORM.)

5. PostgreSQL Official Documentation

   https://www.postgresql.org/docs/

   (Used for designing and managing the relational database structure.)

6. HTML5 and CSS3 References – W3Schools

https://www.w3schools.com/

(Used for creating the front-end layout and styling the web interface.)

7. JavaScript Basics – Mozilla Developer Network (MDN)

   https://developer.mozilla.org/en-US/docs/Web/JavaScript

   (Used for enhancing UI interactivity and handling client-side events.)

8. Python Programming Tutorials – GeeksforGeeks

   https://www.geeksforgeeks.org/python-programming-language/

   (Referred for general Python programming, error handling, and logic structuring.)

9. Git and GitHub Guides – Git Documentation

   https://git-scm.com/doc

   (Used for version control, collaboration, and code backup.)

10. PySide6 (Optional)

    https://doc.qt.io/qtforpython/

    (Used in earlier GUI experiments before migrating fully to web interface.)