

[Click Here](#)



Examples of major and concentration project

What are you going to major in? What's your major? What did you major in? As you are applying to college, attending college, and especially after you graduate college, the most common question others will ask about your undergraduate experience is what you studied and why. While college students take a variety of classes and explore a number of different subjects, each of us chooses just one (or sometimes two) disciplines to focus on or "major in." You might have noticed that Harvard uses the word "Concentration" instead of "Major," but no matter the name, it's the subject you take most of your classes in and which will appear on your degree. If you're just starting your college search process, you'll want to start thinking about what you might want to study. Do you love writing and your high school English classes? Do you prefer mathematics and problem solving? Or do you love building things and figuring out the science behind how they work? It helps to think about what subjects you enjoyed in high school as well as what you're passionate about doing in your free time. While there are some schools that ask you to apply to specific majors, there are many schools (like Harvard) where you can say what you think you might want to study and figure it out once you're there. If your dream is to become a doctor, for example, you'll want to make sure that the colleges you're considering offer pre-med courses and advising. This is why it's good to have a sense of what you might want to study as you're building your college list. I miss doing reading for my English and creative writing classes in my Winthrop House dorm room! Don't know what you want to study yet? Don't worry! I entered college convinced that I would study government and go on to work on political campaigns or go to law school and, within my first three weeks of starting school, I pivoted in the direction of history and sociology. Once I had the opportunity to explore different classes and departments, I realized that I really loved reading and learning about social theory and literature. The idea behind a liberal arts education is that you get to study a lot of one thing (your major) and a little bit about everything else. I concentrate in Social Studies with a secondary (or minor) in English — that means that most of my classes are about social theory, many of my classes are about literature, and I've also taken math, economics, art history, and an assortment of other classes that met Harvard's General Education requirements or just seemed interesting and fit in my schedule. Minors, too, are another way to get in-depth exposure or experience with a subject without committing to majoring in it. They're basically mini-majors — you usually only have to take 4-6 classes in a given subject to earn a minor (or, at Harvard, a secondary) in it. I chose to declare an English secondary because I really enjoy reading literature and talking about it with professors and my classmates. At Harvard, I've been able to use my secondary field to take classes on creative writing, Shakespeare, women's literature, and even the feeling of boredom as it manifests in literature. I like to make sure that I have a balance in my schedule between social science classes for my major, graduation requirements (like science or math), and English classes, which are just for fun! It might feel daunting to have to choose just one thing out of what feels like an infinite number of options for your college degree but, ideally, the subject you choose will be one that you are passionate about. Trust your gut and keep an open mind — whatever you choose to study, make it something you love! College 101 is a recurring series in which Harvard First Generation Program coordinators answer the questions about college you didn't know you needed to ask! For more advice and information about applying to or attending college, you can email HFGP at firstgen@fas.harvard.edu. Next Story: Reflections from a Senior Tour Guide Read Student Voices Read Student Voices Read Student Voices Read Student Voices As cyber threats have reached \$10.5 trillion annually, the demand for professionals in this field grows. This field focuses on developing solutions like firewalls, encryption algorithms, and intrusion detection systems to protect systems, networks, and data. The final year projects ideas below will help you gain the skills and tools needed to secure digital spaces and make a successful career in cybersecurity. 14. User Authentication Software In this project, you'll create a secure system that handles new user registrations, logins, and password resets. If you have a bit of web development knowledge, you'll learn how to protect your application from unauthorized access and keep user data safe. What Will You Learn? Encryption & Hashing: You'll secure passwords and user credentials properly. User Role Management: You'll set different permissions for admins, staff, or end-users. Session Handling: You'll manage tokens or cookies for ongoing user sessions. Database Security: You'll store sensitive info in a way that reduces potential breaches. Tech Stack & Tools You'll Master Languages: Python, JavaScript, or Java Libraries: bcrypt (for password hashing), JWT (for token-based authentication) Frameworks: Django, Flask, or Node.js + Express Database: MySQL or MongoDB Key Project Features Registration with email verification Secure login sessions with token-based authentication Password reset via email or security questions Admin dashboard to manage user roles Real-World Examples/ Uses of The Project Scenario Description E-commerce Platforms Safeguards user accounts and payment details. Internal Company Portals Controls access based on roles or departments. Project Challenges Blocking brute-force and SQL injection attempts Providing a smooth login process without slowing the site Handling logouts, session timeouts, and token refresh Ensuring user data privacy and compliance with local regulations Also Read: What is Cybersecurity? Definition, Types, Career, Job Roles & Salary 15. Face Detector Application In this project, you'll develop a tool to recognize human faces in images or video streams. If you have some knowledge of computer vision libraries, this is a great way to explore real-time detection or authentication systems. What Will You Learn? Computer Vision Basics: You'll identify and locate faces in images or live video. Machine Learning Concepts: You'll understand how trained models detect and classify facial features. Image Preprocessing: You'll handle lighting variations, angles, and image noise. Integration: You'll tie your detection module into applications (like a security door or attendance system). Tech Stack & Tools You'll Master Languages: Python, C++ Libraries: OpenCV, Dlib, or TensorFlow Frameworks: Flask or Django (if you build a web-based interface) Hardware (Optional): Raspberry Pi or other IoT devices for edge detection Key Project Features Real-time face detection from a camera feed Face logging or recording for security checks Optional notification system for unrecognized faces Storage of recognized faces for quick matching Real-World Examples/ Uses of The Project Scenario Description Security Systems Alerts staff when an unknown person enters. Attendance Tracking Automatically logs attendance without manual input. Project Challenges Dealing with low-quality or angled images Reducing false positives in crowded scenes Speed optimization for real-time performance Respecting privacy and data storage rules 16. SMS Spam Filtering In this major project for CSE final year students, you'll create a system that scans incoming text messages and separates spam from legitimate ones. If you have a background in NLP, this is an interesting way to fight unwanted ads, phishing attempts, or malicious links. What Will You Learn? Text Classification: You'll label messages as "spam" or "ham" through machine learning. Feature Extraction: You'll pick out keywords, links, or patterns that commonly appear in spam. Performance Metrics: You'll evaluate how well your model handles false positives/negatives. Real-time Detection: You'll integrate this with a mobile or web interface for instant filtering. Tech Stack & Tools You'll Master Languages: Python, Java Libraries: Scikit-Learn, NLTK, or TensorFlow (for advanced approaches) Frameworks: Flask (if you want a simple web API) Database: SQLite or MongoDB (to store message logs) Key Project Features Automated message scanning in near-real time Classification model that flags suspicious content Logging and reviewing flagged messages User-friendly dashboard to adjust filtering rules Real-World Examples/ Uses of The Project Scenario Description Personal Smartphones Cuts down on distracting spam texts. Call Centers Filters incoming messages for agents. Project Challenges Handling regional languages or slang Balancing sensitivity so valid messages aren't marked as spam Updating the model as new spam techniques emerge Managing large volumes of messages at scale 17. Multiple-factor Authentication System In this project, you'll add layers of security to a standard login — like an OTP on a phone or a biometric scan — so that even if passwords leak, attackers can't get in. It's a strong introduction to creating secure online portals. What Will You Learn? Two-factor or Multifactor Concepts: You'll combine something users know (password), have (phone/OTP), or are (fingerprint). API Integration: You'll work with SMS or email gateways to deliver verification codes. App Security: You'll protect both the user's login and the second layer from attacks. User Experience: You'll keep authentication steps simple, so people don't get frustrated. Tech Stack & Tools You'll Master Languages: Python, JavaScript, or Java Libraries: PyOTP or Twilio (for OTP), Google Authenticator APIs Frameworks: Django, Flask, or Node.js + Express Database: MySQL or PostgreSQL Key Project Features Traditional username/password login Secondary prompts (OTP, push notification, biometrics) Secure sessions with tokens or cookies Admin panel to manage roles and access Real-World Examples/ Uses of The Project Scenario Description Banking Apps Adds OTP or fingerprint scans for secure transactions. Corporate Portals Protects sensitive company data with extra checks. Project Challenges Ensuring seamless flow without irritating users Handling time-based OTPs or sync issues Storing and transmitting biometric data securely Integrating multiple authentication methods for different user profiles Final Year Projects Ideas in IoT & Smart Technologies Think of a world where everyday devices are effortlessly connected, making life more convenient. IoT and Smart Technology projects let you build systems for smart homes, health monitoring, and automation. You'll gain hands-on experience with sensors, data transmission, and cloud connectivity, preparing you to design future technology. Below are some final year project ideas in IoT that will help you develop critical job-ready skills. 18. Surveillance Camera using IoT In this project, you'll create a camera setup that streams live footage to the cloud. You'll combine sensors (for motion detection) with internet connectivity to monitor spaces in real-time. If you have basic IoT and networking skills, you can build a system that helps prevent unauthorized access or quickly detect issues. What Will You Learn? Device Connectivity: You'll link cameras and sensors to a local network or the internet. Cloud Integration: You'll store or stream captured video to AWS or Google Cloud services. Motion Detection: You'll use image processing libraries (like OpenCV) to identify movement. Data Security: You'll ensure feeds are encrypted and accessible only to authorized users. Tech Stack & Tools You'll Master IoT Hardware: Raspberry Pi or ESP32 Cloud Platforms: AWS or Google Cloud for data storage/streaming Libraries: OpenCV (for motion detection) Protocols: MQTT or HTTP for sending alerts Key Project Features Live video streaming accessible from a web/mobile app Automatic alerts or notifications upon motion detection Night vision or low-light support (with the right hardware) Secure login for remote viewing Real-World Examples/ Uses of The Project Scenario Description Home Security Allows homeowners to check live feeds from anywhere. Warehouse Monitoring Tracks product movement and guards against intrusions. Project Challenges Handling network bandwidth limits for video streaming Ensuring stable power and internet connection for 24/7 uptime Minimizing false motion alerts (e.g., from pets or passing vehicles) 19. Wireless Sound Control In this project, you'll design a system to manage audio levels or switch music tracks remotely across multiple rooms or devices. You'll integrate IoT modules so users can control volume and playback without manually adjusting each device. Basic knowledge of embedded systems and networking can set you up for success here. What Will You Learn? Wireless Protocols: You'll explore Bluetooth or Wi-Fi modules for sending commands. Signal Processing Basics: You'll adjust volume, balance, or track controls programmatically. Device Synchronization: You'll handle multiple speakers or devices at once. App Integration: You'll build a simple interface to manage audio settings on the fly. Tech Stack & Tools You'll Master IoT Hardware: ESP8266 or Raspberry Pi Software Tools: Arduino IDE or Python for microcontroller programming Wireless Standards: Bluetooth Low Energy (BLE) or Wi-Fi Framework (Optional): Blynk or a custom mobile app for remote control Key Project Features Centralized volume and track control Option to group devices for synchronized audio User-friendly dashboard on phone or web Potential for scheduling or automated sound levels Real-World Examples/ Uses of The Project Scenario Description Home Theater Systems Manages multiple speakers or smart soundbars at once. Office/Conference Rooms Adjusts presentations and announcements easily. Project Challenges Ensuring low-latency commands for real-time changes Handling network connectivity issues across multiple devices Keeping the audio in sync if multiple speakers play simultaneously Also Read: How Does IoT Work? Top Applications of IoT 20. Smart College Campus Surveillance In this project, you'll combine IoT cameras, sensors, and possibly AI modules to secure a campus. You'll record events, detect suspicious movements, and maybe even integrate facial recognition if you're up for a challenge. This helps colleges automate their security checks and respond quickly to unusual activity. What Will You Learn? Multi-Camera Networking: You'll connect multiple devices across a wide area. Real-Time Monitoring: You'll stream live video or sensor alerts to a central dashboard. Data Analytics (Optional): You'll analyze video or sensor data for patterns (e.g., crowd detection). Scalability: You'll design a system that can expand to new campus areas without major rework. Tech Stack & Tools You'll Master IoT Hardware: Raspberry Pi, Arduino, or IP cameras Networking Protocols: TCP/IP, MQTT for sensor communication Cloud Services: AWS or Azure for data storage and remote monitoring AI/Computer Vision (Optional): OpenCV for advanced detection features Key Project Features Unified dashboard showing multiple camera feeds Automatic alerts for trespassers or odd-hour movements Integration with door locks or alarms for immediate actions Data logging to review past incidents Real-World Examples/ Uses of The Project Scenario Description Large University Campuses Monitors dorms, labs, libraries, and sports areas in real time. Research Facilities Secures high-value equipment with continuous surveillance. Project Challenges Managing data traffic for multiple high-resolution camera streams Ensuring reliable coverage in large or multi-building campuses Balancing privacy with security requirements 21. Smart Farming Using IoT Technologies In this project, you'll use sensors to track soil moisture, temperature, and other conditions in farming. You'll automate irrigation or send alerts to farmers when crops need attention. This is a practical way to improve crop yields and resource usage. What Will You Learn? Sensor Calibration: You'll set up soil moisture or temperature sensors accurately. Data Transmission: You'll send sensor readings to a central server or cloud. Automated Controls: You'll switch on irrigation pumps or fans based on sensor data. Data Visualization: You'll display historical trends so farmers can make better decisions. Tech Stack & Tools You'll Master IoT Hardware: Arduino or Raspberry Pi Sensors: DHT11 (temperature/humidity), soil moisture sensors Cloud Platforms: AWS IoT, Firebase, or Azure for data logging Front-End (Optional): React, Angular, or a simple web dashboard Key Project Features Real-time sensor updates on moisture and temperature Automated watering based on threshold values Alerts via SMS or app notifications Historical data analysis to optimize resource use Real-World Examples/ Uses of The Project Scenario Description Greenhouses Maintains ideal conditions for high-value crops. Open Fields Helps large-scale farms reduce water wastage. Project Challenges Ensuring sensor accuracy over large fields Providing stable connectivity in remote areas Managing power consumption for sensors or controllers Handling unpredictable weather factors Also Read: 6 Best IoT Projects Using Arduino Final Year Projects Ideas in Cloud Computing Cloud computing is transforming how businesses manage data, with market revenue projected to reach a whopping USD 1,806 billion by 2029. Final year project ideas in this field often involve building scalable applications, deploying services, and implementing cloud security protocols like SSL/TLS encryption, identity management, and secure API integrations. Explore the most innovative projects below to learn how to create flexible, reliable, and secure systems that scale globally. 22. Cloud-based File Storage System A setup like this allows you to store and retrieve files from anywhere without relying on physical storage. It's among the most practical final year projects for computer science because it shows how to handle distributed data and ensure secure access. What Will You Learn? Cloud Storage Basics: You'll interact with AWS S3, Azure Blob, or similar services. File Versioning & Sharing: You'll manage file updates and define who can view or edit them. Access Controls: You'll set permissions for different user roles. Scalability: You'll design a system that grows smoothly as file volumes increase. Tech Stack & Tools You'll Master Languages: Python, JavaScript Cloud Platforms: AWS, Azure, or Google Cloud Database (Optional): MongoDB or MySQL Framework (Optional): Flask, Node.js, or Laravel for a file-sharing interface Key Project Features Easy file uploads and downloads Public, private, or team-based file sharing Automatic backups and version history Encrypted storage for added security Real-World Examples/ Uses of The Project Scenario Description Team Document Sharing Multiple users collaborate on shared files. Personal Cloud Storage Individuals keep personal photos or data. Project Challenges Managing large files or high traffic loads Dealing with downtime or migration across cloud providers Ensuring robust authentication and encryption 23. Cloud-based Data Backup Solution A backup system in the cloud helps you regularly copy critical data to remote servers, ensuring you can restore everything if a local drive crashes. This approach is vital for businesses or individuals who cannot afford to lose their data. What Will You Learn? Automated Backup Scheduling: You'll set daily, weekly, or monthly intervals. Data Encryption & Privacy: You'll secure data during transit and storage. Recovery Procedures: You'll restore data with minimal downtime. Monitoring & Alerts: You'll get notifications when backups fail or exceed storage limits. Tech Stack & Tools You'll Master Languages: Python or Bash scripting Cloud Services: AWS, Google Cloud, or Azure Database (Optional): MySQL or PostgreSQL if backing up DBs Backup Tools: rclone, AWS CLI, or custom scripts Key Project Features Incremental backups to save bandwidth Versioning for point-in-time recovery Encrypted data at rest and in transit Dashboard or logs to view backup status Real-World Examples/ Uses of The Project Scenario Description Small Businesses Protect invoices, client data, and internal docs. Personal Backups Keep photos or important files safe from disk errors. Project Challenges Scheduling backups during off-peak hours Preventing partial or corrupted uploads Handling large data sets that grow over time Also Read: Data Security in Cloud Computing; Top 6 Factors To Consider 24. Serverless Web Application A serverless approach removes the need to maintain your own servers. You write functions that run on-demand, paying only for actual usage rather than an always-on system. It's a smart pick if you want a major project for CSE final year that highlights modern cloud architectures. What Will You Learn? Function-as-a-Service (FaaS): You'll deploy code to AWS Lambda, Azure Functions, or GCP Cloud Functions. Event Triggers: You'll respond to events like HTTP requests, database updates, or file uploads. Scalability: You'll rely on the cloud provider to handle traffic spikes automatically. Cost Optimization: You'll pay per execution, avoiding hefty hosting costs for idle time. Tech Stack & Tools You'll Master Languages: Node.js, Python, or Go Cloud Providers: AWS Lambda, Google Cloud Functions, Azure Functions Database (Optional): DynamoDB, Firestore, or SQL-based APIs & Gateways: Amazon API Gateway or equivalent for handling requests Key Project Features Lightweight microservices handling specific tasks Automatic scaling without manual server setup Pay-per-use model with minimal overhead Logging and monitoring integrated into the cloud Real-World Examples/ Uses of The Project Scenario Description Chatbot Services Triggers a function only when a user sends a query. Data Processing Pipelines Processes files on arrival, then stores results. Project Challenges Dealing with "cold starts" (slight delays when functions wake up) Managing limited execution time or memory constraints Coordinating multiple functions to form a cohesive app 25. Cloud Resource Management System An application like this tracks various resources — servers, containers, or databases — in the cloud. While executing this project, you'll automate allocations, control costs, and ensure everything runs smoothly, especially useful for large teams. What Will You Learn? Resource Provisioning: You'll spin up or shut down servers based on user demand. Cost Tracking: You'll monitor how much each resource costs to keep budgets in check. Auto-scaling Logic: You'll design scripts to adjust capacity during traffic spikes. Multi-Cloud Integration (Optional): You'll manage resources across AWS, Azure, or GCP. Tech Stack & Tools You'll Master Languages: Python or JavaScript Cloud APIs: AWS SDK, Azure CLI, or GCP API Infrastructure as Code: Terraform, AWS CloudFormation, or Ansible Databases: MongoDB or SQL for storing resource info Key Project Features Real-time monitoring of CPU, memory, and network usage Automated alerts for threshold breaches Role-based access so teams can only adjust certain resources Detailed billing or usage reports Real-World Examples/ Uses of The Project Scenario Description Cloud Management Large companies track thousands of servers or services. Startup Scaling New ventures scale up or down without spending too much. Project Challenges Handling diverse resources across different cloud providers Avoiding over-provisioning and wasteful spending Keeping systems secure while automating critical operations Want to excel in Cloud Computing? Join upGrad's Cloud Computing Courses and get started now! The projects below have been selected by the MA in Education department as examples of quality capstone projects rather than absolute models of content and format. If you have specific questions concerning the writing and preparation of your capstone project, please contact your instructor. For help researching your topic, please contact a librarian.