

Pimpri Chinchwad Education Trust's

PIMPRI CHINCHWAD COLLEGE OF ENGINEERING

SECTOR NO. 26, PRADHIKARAN, NIGDI, PUNE 411044

An Autonomous Institute Approved by AICTE and Affiliated to SPPU, Pune

DEPARTMENT OF COMPUTER ENGINEERING



Curriculum Structure and Syllabus
of
Minors in Advanced Web Development
(Course 2020)



Effective from Academic Year 2022-23

Institute Vision

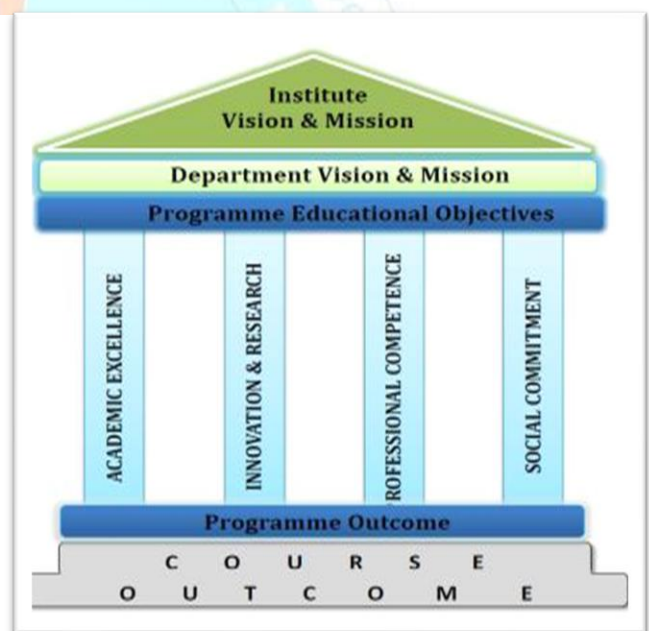
To serve the Society, Industry, and all stake holders through Value-Added Quality Education.

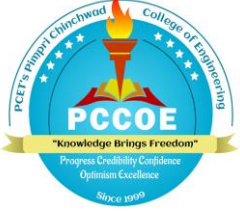

Institute Mission

To serve the need of the society at large by establishing state-of-the-art Engineering, Management and Research institute and impart Attitude, Knowledge and Skills with Quality Education to develop individuals and teams with ability to think and analyze right values and self-reliance.

Quality Policy

We at PCCOE are committed to impart Value Added Quality Education to satisfy the applicable requirements, needs and expectations of the Students and Stakeholders. We shall strive for academic excellence, professional competence and social commitment in fine blend with innovation and research. We shall achieve this by establishing and strengthening state-of-the-art Engineering and Management Institute through continual improvement in effective implementation of Quality Management System.



	Pimpri Chinchwad Education Trust's Pimpri Chinchwad College of Engineering	
Course Approval Summary		

A) Board of study - Department of Computer Engineering

Sr. No.	Name of the Course	Course Code	Page number	Signature and stamp of BoS
1.	Getting started with JavaScript	WD101	5	
2.	Server-side Programming with Node.js	WD201	8	
3.	Front-end Development with React & TypeScript	WD301	11	
4.	Getting Ready for Production	WD401	14	

Approved by Academic Council:

Chairman, Academic Council
Pimpri Chinchwad College of Engineering

Approved by Board of Governors:

Chairman, Board of Governors
Pimpri Chinchwad College of Engineering

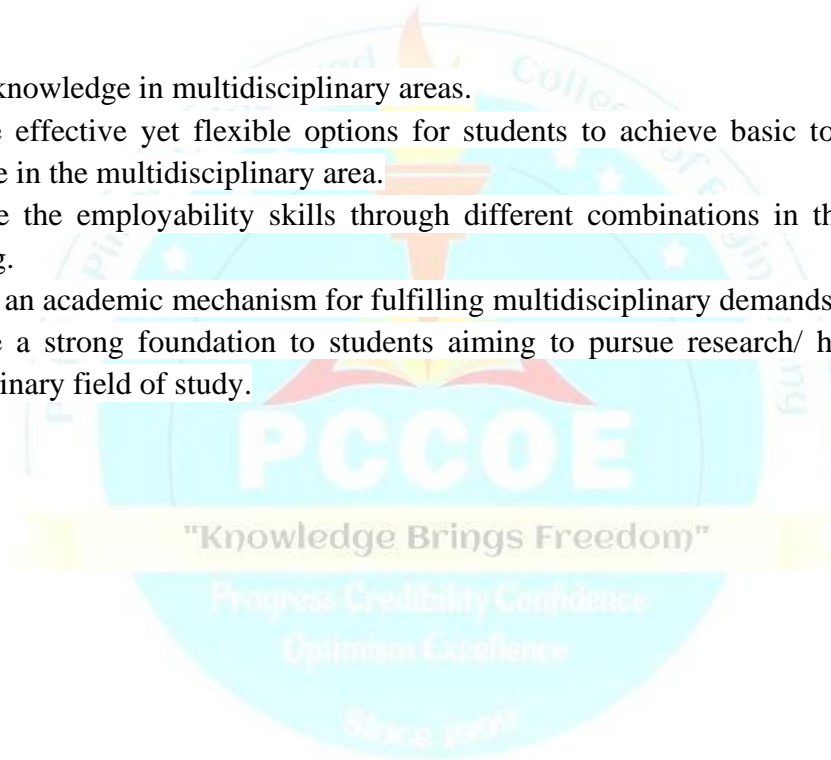
Preface

Looking at Global Scenario to enhance the employability skills and impart deep knowledge in emerging/ multidisciplinary areas, an additional avenue is provided to passionate learners through the Minors and Honors Degree Scheme in academic structure.

For **Minors degree** program, the student has to earn additional 18-20 credits in multidisciplinary areas of other domains.

Objectives of Minors Degree

- To impart knowledge in multidisciplinary areas.
- To provide effective yet flexible options for students to achieve basic to intermediate level competence in the multidisciplinary area.
- To enhance the employability skills through different combinations in the diverse fields of engineering.
- To provide an academic mechanism for fulfilling multidisciplinary demands of industries.
- To provide a strong foundation to students aiming to pursue research/ higher studies in an interdisciplinary field of study.



Preface of Minors in Advanced Web Development

The Minor Degree in Advanced Web Development Curriculum is approved by All India Council for Technical Education (AICTE) under the National Educational Alliance for Technology (NEAT) programme of the Ministry of Education, Government of India. This can be opted for and pursued by students who are enrolled in the Institutes selected as part of LITE programme.

Advanced web development is an industry led Minor degree programme that would enable you to become a skilled web developer in the global Software as a Service (SaaS) Industry. This programme is available for students from Institutes whose faculty coordinator has completed the faculty training requirements. These Institutes are amongst the selected institutions who have qualified for AICTE's Leadership in Teaching Excellence (LITE) program. Within these institutions, students can apply based on the Minor Degree format implemented in their respective Institute.

This course was initially created based on Freshworks' internal training program for their new employees, and is continuously updated to meet students' requirements and changing industry demands. The intent of this course is to create a pathway for students from all engineering disciplines to the global web development industry.

Objective of Minor Degree

This program aims to

- Learn the professional web development knowledge and practical skills used every day in the industry.

Learning Outcomes

At the successful completion of this Minor program, students will be able to

- Set up a web development environment for Node.js and React & TypeScript
- Be able to set up a CI/CD pipeline for a server-side application, ensuring the code reaches production automatically after tests pass.
- Know how to organize & communicate development work using pull requests.
- Be aware of container-based deployments, be able to build a Docker image for their web application, and then deploy that image to a web server.
- Know how to set up a web application to support localization.
- Set up error-logging for their web application to capture runtime errors - both in the back-end and in the front-end. They'll also know how to write tests that replicate errors before implementing a fix to prevent regressions.

INDEX

Sr. No.	Content	Page No.
1	List of Abbreviations in Curriculum Structure	1
2	Curriculum Structure	2
3	Course Syllabus of Semester – V Courses	4
4	Course Syllabus of Semester – VI Courses	7
5	Course Syllabus of Semester – VII Courses	10
6	Course Syllabus of Semester – VIII Courses	13
7	Vision and Mission of Computer Engineering Department	16

LIST OF ABBREVIATIONS IN CURRICULUM STRUCTURE

Sr. No.	Abbreviation	Expansion
1.	L	Lecture
2.	P	Practical
3.	T	Tutorial
4.	H	Hours
5.	CR	Credits

Curriculum Structure

Minors in Advanced Web Development

CURRICULUM STRUCTURE**Structure for Minors in Advanced Web Development
(Computer Engineering)**

Semester	Course Code	Course Name	Teaching Scheme				Cr	Evaluation Scheme
			L	P	T	H		
V	WD101	Getting Started with JavaScript	0	0	1	1	1	The courses shall be continuously evaluated with 50% of the marks for continuous evaluation and 50% of marks for the final capstone project where students demonstrate their knowledge by creating a web application. Due to the continuous evaluation scheme, students who make mistakes shall receive personalized feedback and are encouraged to correct their mistakes and resubmit.
VI	WD201	Server-side Programming with Node.js	0	0	6	6	6	
VII	WD301	Front-end Development with React & TypeScript	0	0	6	6	6	
VIII	WD401	Getting Ready for Production	0	10-14	0	10-14	5-7	
Total			0	10-14	13	23-27	18-20	

L-Lecture, P-Practical, T-Tutorial, H-Hours, Cr-Credits.

Important Note

- The syllabus is approved by AICTE. This is self-paced program. Students need to follow the deadlines as per Pupilfist LMS.
- Students may choose to replace WD 401 with an internship as well.

Course Syllabus
Semester – V
Minors in Advanced Web Development

Program:	Minors in Advanced Web Development			Semester:	V
Course:	Getting Started with Javascript			Code:	WD101
Teaching Scheme				Evaluation Scheme	
Lecture	Tutorial	Credit	Hours	The courses shall be continuously evaluated with 50% of the marks for continuous evaluation and 50% of marks for the final capstone project where students demonstrate their knowledge by creating a web application.	
0	1	1	1		
Prior Knowledge of :					
1. A computer with a modern OS (Windows 10 or above, Ubuntu 20.04 and above, macOS 10.15 and above). It includes skills like the ability to browse the internet and find information, ability to use software applications like word processors and spreadsheets, and be comfortable with user-level operating system concepts like CPU, memory, disks, and files and folders.					
is essential.					
Course Objectives:					
1. This course is meant for students who do not have prior programming experience, or have a light background, and are looking to build a robust foundation for computational thinking. 2. They'll learn to deconstruct what software applications do, and reason about the essence of computation as transformation of data from one shape to another. 3. Practically, they will be able to set up a development environment, be introduced to HTML & CSS, and learn to program in a functional subset of JavaScript. 4. By the end of the course, they will build an Online Registration form that runs on the browser, with the ability to store and retrieve submissions using browser native web storage. 5. They will also be able to create and deploy a simple and basic website to the internet.					
Course Outcomes:					
By completing the WD 101 course, students will gain a foundation in programming and computational thinking, and be introduced to the field of web development. Specifically, they will learn how to:					
1. Set up a development environment. 2. Create and style basic web pages. 3. Transform data with JavaScript. 4. Use computational abstractions. 5. Work with the HTML forms. 6. Work on native HTML Form Validations. 7. Understand Web Storage for saving and retrieving data.					
Detailed Syllabus:					
Unit	Description				Duration(H)
I	Welcome to the course This module introduces students to the World Wide Web. Students are also guided through setting up a development environment on their computer. Students are taught to set up Visual Studio Code as their editor and to use Prettier and ESLint extensions for code formatting and code quality respectively.				

II	<p>Let's create our own websites!</p> <p>In this module students learn how to develop a simple website using HTML. They experiment with some useful html tags, learn how to look inside websites. The students deploy the website they develop and share it over the Internet.</p>	12
III	<p>Basic Introduction to HTML and CSS</p> <p>This module gives some basic introduction of HTML and CSS. Students learn how to put together a web page that contains HTML, CSS, and JavaScript.</p>	
IV	<p>Style Matters</p> <p>This module teaches students how to style web pages using CSS. Students also learn how to use Tailwind CSS to add custom styling to their webpage.</p>	
V	<p>Working with JavaScript data types</p> <p>In this module students are introduced to different data types - Number, Boolean and String. They carry out various operations on these data types to understand the difference between them and also can decide the suitability of a data type given a task or operation.</p>	
VI	<p>Working with JavaScript data structures</p> <p>This module teaches students how to iterate with arrays using forEach method and generate an HTML list from an array. Students perform various transformations on an array using the map method and are introduced to filtering of arrays. Students are also introduced to objects in JavaScript. They learn how to create objects, add and access properties of objects and perform various operations on them.</p>	
VII	<p>Functions - code we can call multiple times</p> <p>This module teaches students how to use functions to modularize the codebase. Students learn how to return values from a function and also how to treat functions as values, by passing them as arguments.</p>	
VIII	<p>Create a form with validations</p> <p>In this module, students learn about HTML form element and form data. They learn how to create a user form, add validations, store and retrieve data. Students develop and deploy their personal website that includes the form they have built with additional validations and display the data submitted by users on the website.</p>	
	Total	
<p>Text/Reference Books:</p> <p>This course does not require students to use physical textbooks. Instead, original course material (videos, text and images) has been prepared for students to go through and is open-sourced under Creative Commons Attribution-ShareAlike 4.0 International License © Freshworks Inc. & Pupilfirst Pvt. Ltd.</p> <p>This course material may include some third-party content with a compatible license, and external links for additional reading on the Internet. Students are also taught how to search for information on their own.</p>		

Course Syllabus
Semester – VI
Minors in Advanced Web Development

Program:		Minors in Advanced Web Development		Semester:	VI	
Course: Server-side Programming with Node.js				Code:	WD201	
Teaching Scheme				Evaluation Scheme		
Lecture	Tutorial	Credit	Hours	The courses shall be continuously evaluated with 50% of the marks for continuous evaluation and 50% of marks for the final capstone project where students demonstrate their knowledge by creating a web application.		
0	6	6	6			
Prior Knowledge of: Web Development 101, before beginning this course. Students should have access to a computer with a modern OS (Windows 10 or above, Ubuntu 20.04 and above, macOS 10.15 and above). is essential.						
Course Objectives: Through the course, the student will work up to build a To-do Management application using Express.js, PostgreSQL, HTML, and CSS. The app will be hosted on the cloud using Heroku. They will then independently work on a capstone project which will be a microcosm of a production web application and the challenges and trade-offs that come with it. Being an industry-led course, the students will also be exposed to professional practices like code reviews, code quality, and version control (git). They'll have access to a Web Development Community where they are encouraged to ask well-crafted and specific questions, a valuable skill in a professional setting.						
Course Outcomes: After learning the course, the students will be able to: <ol style="list-style-type: none"> 1. Build web applications using Express.js. 2. Manipulate data using both imperative and functional programming techniques. 3. Model real-world systems using object-oriented design 4. Write HTML & CSS to create elegant web pages 5. Build database applications using Sequelize. 						
Detailed Syllabus:						
Unit	Description				Duration(H)	
I	Introduction to Node.js In this module students are introduced to Node.js - they learn how to install it and write programs on it and use Node.js REPL. Students also start using GitHub and learn how to collaborate on code with others using the git tool.				72	
II	Working with NPM This module is an introduction to Node.js package manager for students where they start writing custom NPM modules. They also explore and use built-in modules of Node.js					
III	Node.js deep dive In this module students start building their first application and learn how to use closure to emulate private methods.					
IV	Testing In this module students are introduced to testing. They start writing tests for their application, learn how to use Jest to run the tests and pre-commit hooks to run the tests automatically before each commit.					
V	Databases and Sequelize In this module students get to learn about databases and set up a PostgreSQL database. They learn how to connect to a database from a Node.js application and then work on the database by creating Sequelize models to manipulate data.					
VI	Backend Web development with Express.js In this module, students develop their first application and connect it to the PostgreSQL database on their machine, and begin learning the basics of the CRUD pattern by building some additional features to the application that they're working on.					
VII	Add User Interface for To-do Application This module teaches students how to create interfaces for their application. They also practice converting a given visual design into working HTML and CSS.					

VIII	<p>EJS Templating This module teaches touches upon the basics of the MVC pattern, instructing student how to render dynamic data inside their HTML pages using EJS templates. This module also lets the student practice how to deploy their work to a remote server.</p>	
IX	<p>HTML forms to save and accept user inputs This module teaches students how to accept user input on their application via form element in HTML. Students also explore more of the CRUD pattern, moving onto creation of resources using forms, deletion of existing resources, and learn about Cross Site Request Forgery (CSRF) and how authenticity tokens can be used to prevent such attacks. Students are also introduced to APIs.</p>	
X	<p>User Authentication and final wrap-up In this module students dig deeper into Sequelize association, migration and validation. They build a functional user sign-up page, learn about password storage and play around with browser cookies, sessions, user authentication, and related best practices. They also learn to display one-off flash messages.</p>	
	Total	72

Text/Reference Books:

This course does not require students to use physical textbooks. Instead, original course material (videos, text and images) has been prepared for students to go through and is open- sourced under Creative Commons Attribution-ShareAlike 4.0 International License © Freshworks Inc. & Pupilfirst Pvt. Ltd.

This course material may include some third-party content with a compatible license, and external links for additional reading on the Internet. Students are also taught how to search for information on their own.

Course Syllabus
Semester – VII
Minors in Advanced Web Development

Program:	Minors in Advanced Web Development			Semester:	VII
Course:	Front-end Development with React & Type Script			Code	WD301
Teaching Scheme				Evaluation Scheme	
Lecture	Tutorial	Credit	Hours	The courses shall be continuously evaluated with 50% of the marks for continuous evaluation and 50% of marks for the final capstone project where students demonstrate their knowledge by creating a web application.	
0	6	6	6		
Prior Knowledge of: Students should have completed Web Development 201, before beginning this course. Access to a computer with a modern OS (Windows 10 or above, Ubuntu 20.04 and above, macOS 10.15 and above) is essential.					
Course Objectives: <ol style="list-style-type: none"> Understand the basic architecture of front end applications and create web applications using React TypeScript front-end stack. Interaction between a client-side application and server-side app via an API. Industry practices for state management and usage of static types. Best practices with regards to the development of a modern client-side application. Learn to build TypeScript projects from scratch to scale. 					
Course Outcomes: By the end of the course the students will: <ol style="list-style-type: none"> Be able to create Single Page Web Applications (SPA) using React, Typescript and TailwindCSS. Have a solid understanding of static types, and know how to port untyped JavaScript to TypeScript. Learn typed state management that is inline with a backend data model. 					
Detailed Syllabus:					
Unit	Description				Duration(H)
I	React fundamentals This module introduces students to development using TypeScript by setting up a development environment, introducing them to the TypeScript programming language and the React framework, and demonstrates some of the basic concepts that underpin the use of React for building dynamic reactive user interfaces.				72
II	State management This module introduces students to the Hooks feature of React, on the usage of callback functions and how to use them to build dynamic components that maintain an internal state. This module also demonstrates state management by building a form and accepting user input.				
III	A deeper dive into React Hooks This module discusses the common pitfalls of state management, introduces in-browser persistent storage, demonstrates additional standard hooks and the creation and use of custom hooks				
IV	Client-side routing This module covers the concept of client-side routing as a separate behaviour from server-side route management. It demonstrates the various aspects of client-side routing such as the use of path parameters, query parameters, programmatic navigation and the operation of links and URLs that are handled client-side.				
V	Types in depth and Variants This module takes a deeper dive into TypeScript's type system, demonstrating concepts such as function types, custom-defined types, generics, and union types. It also instructs the student why the "any" type should be avoided in practice, and finishes up with a				

	demonstration of TypeScript’s type inference behaviour.	
VI	Modelling and managing complex states This module teaches students how to manage complex states using the state reducer pattern, and then demonstrates the pattern by implementing it using React’s useReducer hook.	
VII	APIs and state modelling Through this module, students are introduced to using APIs to interface their client-side code with the server-side, how to model types to allow this interaction to take place, how to maintain a session with the backend, and how to work with pageable APIs.	
VIII	Best practices and npm packages This module covers the best practices of front-end development, including the importance of accessibility and WAI-ARIA standards, and use of third-party packages from the NodeJS ecosystem.	
IX	Production React Apps This final module focuses on production-specific optimizations of a React application, best practices for its build & deployment process, and the configuration of a progressive web app.	
Total		72

Text/Reference Books:

This course does not require students to use physical textbooks. Instead, original course material (videos, text and images) has been prepared for students to go through and is open-sourced under [Creative Commons Attribution-ShareAlike 4.0 International License](#).

This course material may include some third-party content with a compatible license, and external links for additional reading on the Internet. Students are also taught how to search for information on their own.

Course Syllabus
Semester – VIII
Minors in Advanced Web Development

Program:	Minors in Advanced Web Development			Semester:	VIII
Course:	Getting Ready for Production			Code:	WD401
Teaching Scheme				Evaluation Scheme	
Lecture	Practical	Credit	Hours	The courses shall be continuously evaluated with 50% of the marks for continuous evaluation and 50% of marks for the final capstone project where students demonstrate their knowledge by creating a web application.	
-	10-14	5-7	10-14		
Prior Knowledge of: Web Development 301, before beginning this course. Access to a computer with a modern OS (Windows 10 or above, Ubuntu 20.04 and above, macOS 10.15 and above). is essential.					
Course Objectives: 1. To allow the student to learn more about production-ready deployments.					
Course Outcomes: After learning the course, the students will : 1. Be able to bundle a codebase with non-trivial JS dependencies and code. 2. Know how to differentiate between popular JS flavours and pick one that is suitable for a task. 3. Understand why testing is important, what TDD is, and be able to write both unit and integration tests for Rails applications that use JS in the front-end. 4. Be able to set up a CI/CD pipeline for a server-side application, ensuring the code reaches production automatically after tests pass. 5. Know how to organise & communicate development work using pull requests. 6. Be aware of container-based deployments, be able to build a Docker image for their web application and then deploy that image to a web server. 7. Know how to set up a web application to support localization. 8. Set up error-logging for their web application to capture runtime errors - both in the back-end and in the front-end. They'll also know how to write tests that replicate errors before implementing a fix to prevent regressions.					
Detailed Syllabus:					
Unit	Description				
I	Workflow using pull-requests This module acts as an advanced guide to the usage of git in development teams, where the norm is to develop on branches, perform peer-reviews, and to re-work based on reviews before merging. Since this cycle is most often performed using online tooling that uses pull requests to achieve this workflow, students are taught how to open a pull request, make changes, submit work for review and then update code based on review.				
II	JS Bundling - integration of JS into non-JS backends This module covers the history of why "bundling" as a process exists for the JS ecosystem, the most common bundling tools, and the general methodology. This module also covers the new "import maps" feature that allows for similar capability without the use of a bundling tool.				
III	Compile to JS languages - options & approaches This module covers the reason why languages that compile to JS exist, the different purposes that they serve, and demonstrate a few of the most popular options and the differences between each.				
IV	Testing This module covers the importance of testing, the different approaches to testing such as unit testing, integration testing, and hybrid testing. It should also cover popular libraries that are used to help with testing, and also common pitfalls in the practice of testing and how to avoid them.				
V	CI/CD - Continuous integration & delivery This module teaches students about modern development processes that enable teams to release changes quickly and often, by leading them through the process of setting up an automated system that detects				

	changes to code to run tests and then linking that to the deployment of code that passes its test suite to a remote server.
VI	Application environments This module teaches students about the different environments in which an application is expected to run. This module explains the differences between the environments that a student has already operated in - development, testing & production, and also introduces the concept of a staging environment which acts as a gateway to the production environment.
VII	Containerization This module covers the field of containerization - where complex applications are packaged to run in isolated spaces called containers. The approach for covering this topic involves the use of the popular Docker (OCI) standard, teaching students how to build a Docker image for their web application, and how to deploy this image to different targets.
VIII	Internationalisation and localisation This module covers i18n, teaching students the basics of setting up their web applications to support users who prefer or require a language different from the default language of the app, and/or live in a timezone that is different from the default. This module also covers L10n, teaching students how to use the i18n framework to customise their web application for another locale.
IX	Error logging & debugging This module covers the practice of logging and notification of runtime errors that occur on a deployed application. This module also covers the process that is followed to detect the source of a bug, and how testing can be used to ensure a fix and to prevent recurrences.
<p>Text/Reference Books:</p> <p>This course does not require students to use physical textbooks. Instead, original course material (videos, text and images) has been prepared for students to go through and is open-sourced under Creative Commons Attribution-ShareAlike 4.0 International License.</p> <p>This course material may include some third-party content with a compatible license, and external links for additional reading on the Internet. Students are also taught how to search for information on their own.</p>	

* **Students may choose to replace WD 401 with an internship as well.**

Vision and Mission of Computer Department

Department Vision

To be a Premier Hub in Computer Engineering in Education and Research.

Department Mission

To build technologically competent and ethically strong individuals for serving the needs of industry and society by providing state-of-the-art resources, opportunities for Learning and Research in Computer Engineering.