

Global Developer Corps (GDC) Artificial Intelligence Workforce Scheme Document 2024-2029

1. Foreword

AICTE, along with Pupilfirst played a pioneering role in creating the United Nations recognized Digital Public Good “CARE” by initiating the Global Developer Corps (GDC) programme to identify and train the brightest engineering students. AICTE GDC fellows from the batch 2021 created the Oxygen Dashboards for COVID-19 War Rooms during the delta wave crisis of May 2021. The batch 2022 expanded the capabilities to create the TeleICU systems now deployed in 206 district hospitals across nine states.

With the advent of ChatGPT, a new technological era in artificial intelligence (AI) has started. The batch 2023 GDC Fellows worked with Open AI and industry partners to upgrade CARE with AI features that will transform the delivery of healthcare services to citizens. The United Nations has recognised the work done by GDC Fellows as the world's 50th Digital Public Good (DPG).

The National Health Mission has requested to scale up the fellowships to support the rollout of CARE as a Palliative Care Grid, showing the large-scale demand for students with AI skill sets. The success of the GDC Fellowships has opened up the potential to scale this model into a GDC AI Workforce scheme for creating an AI workforce of 100,000 highly skilled engineers over five years, who can work as nation builders and bring the transformative effects of the AI revolution in society.

2. Scheme Objectives:

The GDC AI Workforce Scheme has two objectives set for achievement over 5 years:

- Create a workforce of **One Lakh** skilled in working with AI to create and upgrade Digital Public Infrastructure and Goods.
 - Build capabilities to transform aspects of the teaching-learning process with Artificial Intelligence to enable scale and relevance to the future.
-

2.1 Eligibility

All Students of AICTE-approved institutions in Engineering and Technology and this scheme is totally free of cost.

2.2 Pedagogy and Assessments

Content based on AICTE’s model curriculum following NEP Learner-centered pedagogy, with self-paced learning via video tutorials, real-world applications as capstone projects, and continuous and personalized assessments.

2.3 Internships

After completing the course, students shall be provided with opportunities to work at organizations like the e-Governance Foundation, National Health Mission, and other identified partners for at least one semester as full-time interns on remote/physical mode.

2.4 Certificates

Students who complete the training and internship programme will receive a GDC AI Workforce Fellow Certificate. {Draft Format in Annexure - 1}

2.5 Credits

Pupilfirst will pursue to issue credits for courses and internships based on the National Credit Framework (NCrF) from the onboarded partner institution(s)/Universities.

3. Fellowship Selection and Training process

A four-stage process covers the various stages, from Applications to AI Grading, Testing and Screening, Selection and Internships with Industry.

Stage 1 – Submission of Application

All students interested in participating shall apply through the online portal set up by Pupilfirst.

Stage 2 - AI Grading and Personal Interview

AI-based tests built on GPT 4 would create personalized questions based on the student's resume and cover foundational programming questions and logical reasoning. All students who score above a defined threshold will be selected for interviews to secure admission into the Fellowship Track.

Stage 3: Selection into Fellowship Track

Students who pass the AI test and interviews will be selected for the Fellowship Track.

Those students who complete both the curriculum training and finish the six-month internships will qualify for the GDC AI Workforce Fellow Certificate.

The fellowship track shall admit highly motivated students and must be completed within 14 weeks.

Stage 4: Internships

The brightest students from the fellowship track will be offered a six-month digital internship with GDC partner organizations, such as the eGovernance Foundation or the National Health Mission, to build and upgrade Digital Public Infrastructure and Goods with AI Features.

4. Roles and Responsibilities

4.1 AICTE

Issuance of circulars and notifications as and when deemed to be to sensitise the stakeholders regarding the GDC scheme and calendar of events.

4.2 PupilFirst

Maintenance of portal pertaining to student registration, conduct of tests and interviews, monitoring of student learning and its assessment, and coordination with external organizations for internships.

5. Administrative Guidelines

5.1. Mission Mode Implementation

NEP 2020 outlines the need to identify *disruptive technologies* that could change the education paradigm. As artificial intelligence is experiencing explosive growth, AICTE shall encourage the GDC scheme.

- a. **Students from All Branches:** Students from all branches of study across all tiers are eligible to apply.
- b. **Integration with Swayam:** Pupilfirst shall work with AICTE Swayam team to integrate basic and introductory courses with the Swayam platform.
- c. **Mission Dashboard:** Pupilfirst shall provide AICTE with a mission dashboard with the details of all applicants and their progress status across states and institutions.
- d. **Increasing Industry Partners:** AICTE and Pupilfirst shall work together to increase the number of organisations growing internships.

5.2 GDC Industry Advisory Board

An eight-member interdisciplinary team comprising leaders with experience in building and scaling Digital Public Goods and Infrastructure shall work together under the leadership of the Chairman, AICTE, in the following roles.

S1	Name, Organization and Credential	Role
1	Prof. T. G. Sitharam Chairman, AICTE	Setting Goals and Targets.
2	Srikanth Nadhamuni, eGovernance Foundation Founding CTO of Aadhaar	Chief Technology Mentor, GDC AI Workforce
3	Pramod Varma, EkStep Foundation Chief Architect, Aadhaar and India Stack DPI	Chief DPI Mentor, GDC AI Workforce
4	Sridhar Vembu, ZOHOO Corporation Member, National Security Advisory Group	Chief Security Mentor, GDC AI Workforce
5	Sandeep Singhal, Venture Philanthropist Board Member, ACT Grants	Chief Philanthropy Mentor GDC AI Workforce
6	Narayan Ramaswamy, SGM ADNOC and Director General, Bharat Tech Foundation	Chief Industry Connect GDC AI Workforce
7	Dr Leena Chandran-Wadia, TransDisciplinary University Member, Drafting Committee of National Educational Policy	Chief Researcher GDC AI Workforce
8	Sanjay Vijayakumar, Co-founder Pupilfirst	Lead on-ground execution

*New members may be added as required by the competent authority.

6. Span of Activity and Key Outcomes Expected

Span of Activity:

The scheme would be implemented over five years from 2024-2029

Key Outcomes Expected

1. Create AI Workforce of One Lakh Students trained in full stack development and AI skillset
2. Showcase NEP’s core principles of learner-centred pedagogy and continuous and personalised assessments by using AI and utilizing platforms such as ABC/APAR/NDEAR.
3. Make a Societal impact by enabling the National Health Mission and other partners to deploy the United Nations-approved Digital Public Good, “CARE,” across all 35 States and UTs.
4. Elevate India’s software power by taking DPGs globally to partner countries.

Note: The scheme shall be run on mission mode with online reviews during the first week of each month by the Competent Authority and GDC Industry Advisory Group to ensure that the roadmap for execution is aligned with the NEP and Industry Developments in AI.

Annexure 1 - Sample of GDC AI Workforce Fellow Certificate



Annexure 2 – Curriculum

Detailed Syllabus

Course Code	:	AI-powered Web Development 101
Course Title	:	Getting Started with JavaScript
Number of Credits & Hours	:	1 (L: 0; T: 1; P: 0) - 15 Hours
Course Category	:	Web Development (WD)

Course Objective:

This course is meant for students who have no prior programming experience or a light background and are looking to build a robust foundation for computational thinking and the use of AI.

They'll learn to deconstruct what software applications do, and reason about the essence of computation as the transformation of data from one shape to another.

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. Additionally, they will gain an understanding of the basics of Artificial Intelligence and how AI-powered tools can assist in web development.

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. They will also be able to create and deploy a simple and basic website to the internet. Furthermore, students will integrate AI tools into their projects, enhancing their coding efficiency and productivity.

Prerequisites:

- Students should have access to a computer with a modern OS (Windows 11 or above, Ubuntu 22.04 and above, macOS 10.15 and above).

- Students should have computer literacy. 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.

Course Contents:

Module 1: Welcome to AI-powered web-development!

In this module, students are guided through setting up a development environment on their computer. They will set up Visual Studio Code as their editor and learn to use Prettier and ESLint extensions for code formatting and code quality, respectively. Additionally, students will install and configure GitHub Copilot to introduce AI-powered coding assistance right from the start.

Module 2: Creating our own websites

In this module, students are introduced to the World Wide Web. They 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.

Module 3: Basic Introduction to HTML and CSS

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.

Module 4: Style Matters

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 webpages.

Module 5: Working with JavaScript data types

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.

Module 6 - Working with JavaScript data structures

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.

Module 7 - Functions - code we can call multiple times

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.

Module 8 - Create a form with validations

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.

Module 9: Introduction to AI for Web Developers

In this module, students delve deeper into the basics of Artificial Intelligence and its broader impact on web development beyond just coding assistance. They will explore the integration of AI into web projects, focusing on practical applications and ethical considerations.

Text/Reference Books (if any):

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](#) ©Pupilfirst Pvt. Ltd. Parts of this course are sourced from an earlier version of this course that is also licensed under [CC BY-SA 4.0](#). That original work is © 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 Outcomes:

By completing the WD 101 course, students will gain a foundation in programming and computational thinking, be introduced to the field of web development, and gain a basic understanding of Artificial Intelligence and its applications.

Specifically, they will learn how to:

- Set up a development environment.
- Create and style basic web pages.
- Transform data with JavaScript.
- Use computational abstractions.
- Work with HTML forms.
- Implement native HTML form validations.
- Understand web storage for saving and retrieving data.
- Integrate AI-powered tools like GitHub Copilot into their development workflow.
- Enhance their coding efficiency and productivity using AI suggestions.
- Recognize the ethical considerations and future trends in AI and web development.

**

Course Code	:	AI-powered Web Development 201
Course Title	:	Server-side programming with Node.js
Number of Credits & Hours	:	6 (L: 0; T: 6; P: 0) 90 Hours
Course Category	:	Web Development (WD)

Course Objective:

The objective of this course is to teach students how to build web applications using the Express.js framework, with a focus on industry-practices like functional programming, object-oriented design, programming style guides, security, and version control. They will also explore the addition of AI-powered features into their web application.

Prerequisites:

Students should have completed *Web Development 101*, before beginning this course. Students should have access to a computer with a modern OS (Windows 11 or above, Ubuntu 22.04 and above, macOS 10.15 and above).

Overview:

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 a cloud deployment platform.

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 the web

development Community on the Discord server, where they are encouraged to ask well-crafted and specific questions, a valuable skill in a professional setting.

Course Content:

Module 1 - 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.

Module 2 - 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

Module 3 - Node.js deep dive

In this module, students start building their first application and learn how to use closure to emulate private methods.

Module 4 - 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.

Module 5 - 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.

Module 6 - 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.

Module 7 - EJS Templating

This module teaches touches upon the basics of the MVC pattern, instructing students 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.

Module 8 - 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.

Module 9 - 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.

Module 10 - 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.

Module 11: Generative AI and web development

In this module, students explore the integration of generative AI, focusing on using ChatGPT to interpret natural language input. They will learn how to convert natural language into structured data and implement it in a web application. Additionally, the module will cover prompt engineering to optimize interactions with generative AI models.

Text/Reference Books (if any):

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](#) ©Pupilfirst Pvt. Ltd. Parts of this course are sourced from an earlier version of this course that is also licensed under [CC BY-SA 4.0](#) and that original version of the work is © 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 Outcomes:

By the end of the course, the students will be able to:

- Build web applications using Express.js.
- Manipulate data using both imperative and functional programming techniques.
- Model real-world systems using object-oriented design
- Write HTML & CSS to create elegant web pages
- Build database applications using Sequelize.
- Be able to integrate generative AI features into their project.

The students would have built fundamental first-principles based knowledge about web development and the practical chops to use them to build real-world software. They would also have learnt what it is to work in a professional software environment, helping build a strong foundation for their transition to the industry as competent professionals.

Course Code	:	AI powered Web Development 301
Course Title	:	Front-end development with React & TypeScript
Number of Credits & Hours	:	6 (L: 0; T: 6; P: 0) - 90 Hours
Course Category	:	Web Development (WD)

Course objective

The course aims at training students on the following fronts:

- Understand the basic architecture of front end applications and create webapplications using React & TypeScript.
- Industry practices for state management and using static types. Learn to port untyped JavaScript to TypeScript.
- Learn to build React projects from scratch to scale.
- Learn the best practices followed in industry for production grade applications.

Prerequisites

Students should have completed *Web Development 201*, before beginning this course. Students should have access to a computer with a modern OS (Windows 11 or above, Ubuntu 22.04 and above, macOS 10.15 and above).

Course outline

Module 1: React fundamentals

This module introduces students to the React framework, and demonstrates some of the basic concepts that underpin the use of React for building dynamic reactive user interfaces. Students bootstrap a React project using Vite.

Module 2: Introduction to TypeScript

This module introduces students are introduced to the TypeScript programming language. They learn how to migrate a React application to TypeScript.

Module 3: State management and handling user inputs with forms

This module introduces students to React States and React component lifecycle. They learn state management by building a form and accepting user input.

Module 4: 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.

Module 5: Routing

In this module, students learn to use react-router to navigate between different static and dynamic pages. They will also learn to deploy a React app to Netlify. It demonstrates the various aspects of client-side routing such as the use of path parameters, query parameters, and programmatic navigation.

Module 6: Making API Request with React

Through this module, students learn to make calls to the external APIs to integrate with the React application they have developed. They also learn to implement the user authentication and how to handle user sessions.

Module 7: React Context and Application state management

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 useReducerhook.

Module 8: Handling advance UI events

Through this module, students will learn how to implement complex UI events in React.

Module 9: Creating Production-Ready React Apps

This final module focuses on production-specific optimizations of a React application, bestpractices for its build & deployment process, and the configuration of a progressive web app.

Module 10: Integrating AI in React Development

In this module, students learn about advanced AI techniques and their applications in front-end development. They will explore how AI can enhance user experiences and optimize web application performance. The module also covers several practical examples of AI-powered features in React applications.

Text/Reference Books (if any):

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](https://creativecommons.org/licenses/by-sa/4.0/) ©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 outcome

By the end of the course the students will:

- Be able to create Single Page Applications (SPA) using React.
- Be able to build robust UIs using the statically typed programming language TypeScript.
- Get better at styling web applications using Tailwind CSS.
- Learn typed state management that is inline with the backend data model.

**

Course Code	:	AI powered Web Development 401
Course Title	:	Getting ready for production
Number of Credits & Hours	:	7 (L: 0; T: 0; P: 14) - 210 Hours
Course Category	:	Web Development (WD)

Course objective

The objective of WD401 is to allow the student to learn more about production-ready deployments.

This can be achieved either via the WD401 course material or through an internship at a company. To complete WD401 by following Pupilfirst's course material, students will need to deploy an application of their choice that integrates learnings from earlier courses, and also work through the material of the course, integrating the new production-readiness concepts that are presented here.

The second option is for the student to gain an internship at a company where they get to work on an application that tests a similar skill-set.

Prerequisites

- Students should have completed *Web Development 301*, before beginning this course.
- Students should have access to a computer with a modern OS (Windows 11 or above, Ubuntu 22.04 and above, macOS 10.15 and above).

Course outline

Module 1: 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.

Module 2: 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.

Module 3: 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.

Module 4: 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.

Module 5: 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.

Module 6: 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.

Module 7: 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.

Module 8: Internationalization and localization

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.

Module 9: 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.

Text/Reference Books (if any):

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](#) ©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 outcome

By the end of the course the students will:

- Be able to bundle a codebase with non-trivial JS dependencies and code.
- Know how to differentiate between popular JS flavours and pick one that is suitable for a task.
- 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.
- 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 organise & 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 localisation.
- 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.