A Brief Summary of the Project/Work to be done

Project Title: College Website

Team Members

Parampreet Singh

Manroop Singh

Dhyanesh Bawa

Objective of Application:

- Better User Interface
- Fast and efficient response time
- Improved user interactivity

Need for the Application:

- Present images of college events to users
- Present all courses available in campus
- Present information about all departments along with teachers
- Present various information on College Committee and Principal
- Provide description about college as a whole

Introduction to Existing System

Current system, Multiple page Web Application works on the top of ASP.NET with manually coded CSS.

Drawbacks of Existing System

- Do not provides any Search Engine Optimization.
- Is not responsive to small screen sizes, example Mobile Phones and tablets.
- Data Inefficient, each page had many similar parts example navigation bar and footer that are separately retrieved for each page.
- Slow as each page of Web Application is loaded from scratch.
- Inconsistence font size, image alignment, navigation buttons placement and many content overflows.

Advantages of Proposed System

- Provides better Search Engine Optimization.
- Responsive to approximately all screen sizes, including Mobile Phones and tablets.
- Data efficient, similar parts example navigation bar and footer are only retrieved once from server for every page of Web Application.
- Fast because only parts of the existing page are changed dynamically. Entire page is not loaded from scratch.
- Better font sizes, image alignments and navigation buttons placements.
- Better modularity and arrangement of front-end code and back-end code.
- Will try to cache items which are time consuming, so that next time site loads faster.

- Better user friendly experience and appearance.

Application Environment

- Production Environment
- Hardware Required

Components	Minimum Requirements (Deployment Server)
Architecture	x86 / x86_64
Processor	Intel Pentium Processor 2, 2.4GHz or equivalent
RAM	512 MB
HDD	20 GB
Number of Servers	2 (front-end and back-end)

Components	Minimum Requirements (End User PC)
Architecture	Any, which supports an Internet Browser (Almost all)
Processor	Intel Pentium Processor 2, 2.4GHz or equivalent
RAM	512 MB
HDD	60 MB

- Software Required

Components	Minimum Requirements (Deployment Server)
Operating Systems	Linux Server / Windows Server
Other software	Node, Node Package Manager
Web Server	512 MB
Browser	Not Needed

Components	Minimum Requirements (End User PC)
Operating Systems	Any OS which support Internet Browser (Almost All)
Browser	Firefox / Chrome / Firefox Based / Chrome Based

Development Tools

- Integrated Development Environment - Visual Studio Code
- CSS Framework - Tailwind
- Browser - Chrome / Chrome-based
- Framework - Node (Fastify for back-end, React for front-end)
- Database - SQLite3
- Database ORM - Sequelize
- Node Package Managers - npm and yarn

- Hosting Providers - Vercel for fronend, Heroku for backend
- Collaboration System - Github

Future Scope and Enhancements

- Many pages can be shifted to back-end server from front-end server for making things more dynamic.
- Students can be given student ID to login to website.
- Published articles can have like and dislike button.
- User Interface on dashboard can be improved.
- Dashboard can let user do more customization.
- Some AI can be added to client side bot.
- Notifications, Quick Links and Events can be added / removed from Telegram bot.
- User interface can be customized according to current trends and features.