**WEB DEVELOPMENT**

**Report submitted as part of the internship program requirement for the degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE ENGINEERING (AI & ML)**

**Submitted by**

**KORADA NIVEDITHA (A23126552026)**

****

**DEPARTMENT OF CSE (AI & ML , DS)**

**ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES (UGC AUTONOMOUS)**

**(Permanently Affiliated to AU, Approved by AICTE, and Accredited by NBA &NAAC with an ‘A+’ Grade )**

**SANGIVALASA, Bheemili Mandal, VISAKHAPATNAM – 531162**

**2023-2027**

**ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES (Affiliated to Andhra University)**

**SANGIVALASA, VISAKHAPATNAM – 5351162**

**2023-2027**

****

**BONAFIDE CERTIFICATE**

This is to certify that this Internship Report “WEB DEVELOPMENT” from CODEC TECHNOLOGIES is the bonafide work of KORADA NIVEDITHA (A23126552026) of III/IV CSM carried out Internship under my supervision.

##### INTERNSHIP REVIEWER CLASS TEACHER

Ms. M.V.Bhuvaneswari Mr. M.Sundar Babu

(AssistantProfessor) (AssistantProfessor)

Department of CSE(AI & ML,DS) Department of CSE(AI & ML,DS)

ANITS ANITS

**HEAD OF THE DEPARTMENT**

DR. K. SELVANI DEEPTHI

Department of CSE (AI & ML,DS)

ANITS

## ACKNOWLEDGEMENT

An endeavor that spans a significant period becomes a success with the advice, encouragement, and support of many well-wishers. I take this opportunity to express my sincere gratitude and appreciation to all those who have been instrumental in making this internship experience both enriching and rewarding.

First and foremost, I extend my heartfelt thanks to **Dr. K.S. Deepthi**, Head of the Department of Computer Science & Engineering (AI & ML, DS) at ANITS, for her invaluable guidance, support, and encouragement throughout this internship. Her mentorship and insights have been crucial to my growth during this period.

I would also like to express my deepest appreciation to **CODEC TECHNOLOGIES** for offering me the opportunity to undertake this internship. I am incredibly grateful to my supervisors and team members, whose continuous guidance, expertise, and support have helped me navigate challenges and enhance my skills.

My sincere thanks go to all the faculty members of the **Computer Science & Engineering (AI & ML, DS) department** for their valuable advice and encouragement. I am equally grateful to the support staff, whose assistance in providing resources whenever required was instrumental in the successful completion of my internship.

**KORADA NIVEDITHA**

**A23126552026**

# CONTENTS

## INTERNSHIP SUMMARY

* + Introduction to HTML
  + Introduction to CSS
  + Introduction to Javascript
  + Introduction to Node.js
  + Introduction to Express.js

## TECHNOLOGY USED

* + Frontend: HTML**,**CSS**,**JavaScript
  + Backend: Node.js –Server-sideruntime,Express.js –Routing&middleware
  + Database: MongoDB –NoSQLdatabase,Mongoose–MongoDB ODM

## PROJECT

* + Blog website

1. **ADVANTAGES AND DISADVANTAGES**
2. **CONCLUSION**
3. **CERTIFICATE**
4. **REFERENCES**
5. **INTERNSHIPSUMMARY**

### IntroductiontoHTML:

**WhatisHTML?**

HTML (HyperText Markup Language) is the standard markup language used to create and structure contentontheweb.Itformsthebackboneofallwebsitesandisresponsiblefordefiningtheelements that appear on a webpage—such as text, images, links, forms, and layout components.

HTMLisnotaprogramminglanguage;it’samarkuplanguage,meaningitannotatescontenttotellthe browser how to display it. HTML documents are made up of elements represented by tags that wrap around content to give it meaning and structure.

### BasicStructureofanHTML Document

AsimpleHTMLfiletypically includesthe following components:

<!DOCTYPEhtml>

<html>

<head>

<title>MyFirstWeb Page</title>

</head>

<body>

<h1>Hello,World!</h1>

<p>Thisisa paragraph.</p>

</body>

</html>

* **<!DOCTYPEhtml>:**DeclaresthedocumentasHTML5.
* **<html>:**Roottagthatwrapsallcontent.
* **<head>:**Containsmetadata(title,linkstoCSS, etc.).
* **<body>:**Containscontentvisibleonthe webpage.

### CommonHTML Tags

* **Headings:**<h1>to<h6> fordefiningtitlesand subheadings.
* **Paragraph:**<p>defines aparagraphblock.
* **Links:**<ahref="URL">Link</a>createsa hyperlink.
* **Images:**<imgsrc="image.jpg"alt="Description">embedsanimage.

##### Lists:

* + <ul>(unordered),
  + <ol>(ordered),and
  + <li>(list item).
* **Tables:**<table>,<tr>(row),<td>(cell).
* **Forms:**Usedforinput,containingtagslike<input>,<textarea>,<button>,and<select>.

### ImportanceofHTMLinWebDevelopment

HTMLplaysacrucialroleinthe**frontend** ofeverywebapplication.Itis:

* The**foundation**forbuildingwebpages.
* Responsiblefor**structuringcontent**anddefining**semantics**.
* Abaseonwhich**CSS** (forstyling)and**JavaScript**(forbehavior)buildon.
* Thefirststepinany**webdevelopmentlearningpath**.

### SemanticHTML

**SemanticHTML**refers tousingHTMLtagsthatclearlydescribetheirmeaning.For example:

* <header>,<footer>,<article>,<section>,and<nav>addclaritytothewebpagelayoutand improve accessibility and SEO.

Usingsemantictagsmakesthecodeeasiertoreadandhelpssearchenginesandassistivetechnologies understand the structure of a webpage.

### Conclusion

HTMListhefirstandmostessentialstepinweb development.Itallowsdevelopersto:

* Definecontentstructure
* Embedmultimedia
* Createinteractiveforms
* Layoutcontentinauser-friendly way

MasteringHTMLprovidesastrongfoundationfordivingdeeperintowebtechnologieslikeCSS, JavaScript, and backend frameworks.

### IntroductiontoCSS

##### WhatisCSS?

**CSS(CascadingStyleSheets)**isastylesheetlanguageusedtocontrolthe**presentationandlayout**of HTML elements on a webpage. While HTML structures the content, CSS styles it—defining how elements should look, including their colors, fonts, spacing, positioning, and responsiveness.

CSShelpsdeveloperscreatevisuallyappealing,consistent,andaccessibleuserinterfacesacrossall devices and screen sizes.

##### WhyUseCSS?

* Separates**content(HTML)**from**design (CSS)**.
* Makeswebsiteseasiertomaintainand scale.
* Enhancesthe**visualuser experience**.
* Enablesresponsivedesignacrossdevices(desktops,tablets, mobiles).

##### TypesofCSS

1. **InlineCSS**

Styleiswrittendirectly insidean HTMLelement usingthe styleattribute.

##### Example:

<pstyle="color:blue;">Thisisablueparagraph.</p>

##### InternalCSS

Writteninsidea<style>tag within the<head>section ofthe HTML document.

##### Example:

<style>h1 {

color:green;

}

</style>

##### ExternalCSS

Stylesarewritteninaseparate.cssfileandlinkedtotheHTMLfileusing <link>.

<linkrel="stylesheet" href="styles.css">

**ExternalCSS** isthemostwidelyusedandscalableapproach.

### BasicCSS Syntax

selector { property:value;

}

##### Example:

body {

background-color: #f0f0f0; font-family:Arial,sans-serif;

}

* **Selector**:TargetstheHTMLelement(e.g.,body,h1,.class,#id)
* **Property**:Definesthestyle(e.g.,color,margin,padding)
* **Value**:Specifieshowthe propertyshouldbeapplied.

##### CommonlyUsedCSS Properties

**Property Description**

color Textcolor

background Backgroundcolor/image

font-size Sizeof thefont

margin Spaceoutsidean element

padding Spaceinsidean element

border Addsaborderaround anelement

display Controlsboxmodel behavior

position Specifieselement positioning

flex,grid Layoutmodelsforcomplex structures

### AdvancedCSSConcepts

Aswebsitesgrowincomplexity,CSSofferspowerfulcapabilitiestomanagedesigninamorestructured and efficient way:

### Selectors

SelectorsinCSSareusedtotargetspecificelementsonawebpagetoapplystyles.Thesecanbebased on tags, classes, or unique IDs. Grouping and combining selectors allows developers to write clean, reusable styles that apply consistently across multiple elements.

### Pseudo-classes

Pseudo-classesenablestylingbasedonuseractionsorelementstates.Forexample,alinkmaychange its color when a user hovers over it. This improves interactivity and feedback, helping users navigate and engage with the interface intuitively.

### Positioning

CSS provides different positioning schemes to control where elements appear on the page. Using propertieslikerelative,absolute,andfixedpositioning,developerscandesignflexibleandlayered layouts. This is particularly useful for elements like headers, buttons, banners, or floating panels.

### BoxModel

Every HTML element is treated as a box in CSS. The box model consists of the content area, padding (spaceinsidethebox),border(edgearoundthebox),andmargin(spaceoutsidethebox).Understanding the box model is key to properly managing spacing and layout on web pages.

### ResponsiveWebDesign

Modernwebsitesmustworkwellonvariousscreensizes—fromlargedesktopmonitorstomobile phones. CSS enables this through responsive design techniques:

### MediaQueries

Mediaqueriesallowstylestochangebasedonthescreenwidth,height,ordevicetype.Thisensuresthat the layout, font sizes, and images adjust dynamically, making the website accessible and readable on all devices.

### LayoutSystems(FlexboxandGrid)

CSS provides advanced layout models like Flexbox and Grid. Flexbox is used foraligning items in a singledirection(roworcolumn),whileGridisidealfordesigningcomplextwo-dimensionallayouts. These systems help build flexible, responsive layouts with minimal code and maximum control.

### EnhancingUserExperiencewithCSS

CSS plays a vital role in shaping how users interact with and perceive a website. A well-designed interfaceimprovesreadability,accessibility,andnavigation.Colors,spacing,animations,andresponsive behavior all contribute to a smoother and more enjoyable user experience.

In this project, CSS helped categorize content visually, create an organized layout for blog posts, and ensurethattheplatformlookedconsistentacrossdevices.Featureslikehovereffects,spacingbetween elements, and mobile-friendly design enhanced the professionalism and usability of the site.

### Conclusion

CSSisessentialforbuildingmodern,attractivewebsites.Itallows:

* Fullcontrolover appearance
* Separationofdesignfrom content
* Responsivedesignforalldevices

MasteringCSSempowersdeveloperstobuildprofessionalanduser-friendlyinterfaces,makingita crucial part of the frontend development skill set.

### InroductiontoJavascript

**WhatisJavaScript?**

**JavaScript** is a high-level, interpreted programming language primarily used to make web pages interactiveanddynamic.Itisoneofthecoretechnologiesofwebdevelopment,alongside **HTML**and **CSS**. While HTML structures content and CSS styles it, **JavaScript adds behavior**—such as respondingtouseractions,updatingcontentdynamically,andenablinganimationsordataprocessing. JavaScript runs directly in the browser, making it **client-side** by default, though modern applications also use it on the **server-side** (e.g., with Node.js).

### RoleofJavaScriptinWebDevelopment

JavaScriptempowerswebpagestorespondtouserinputwithoutneedingtoreloadthepage.Itsmain roles include:

* **FormValidation**:Ensuringinputfieldsarefilled correctlybeforesubmission.
* **EventHandling**:Reactingtoclicks,keypresses,andotheruser actions.
* **DynamicContentUpdate**:ModifyingHTML/CSSinreal-time(e.g.,showing/hidingelements).
* **InteractiveUIElements**:Creatingdropdowns,sliders,modals,etc.
* **APICommunication**:Sendingandreceivingdatausingtechnologieslike AJAXorFetch.

### JavaScriptintheBrowser

Webbrowserscome withbuilt-inJavaScriptengines(e.g.,V8inChrome,SpiderMonkeyinFirefox) thatinterpretandrunJavaScriptcode.Thisallowsdeveloperstoenhancewebsiteswithoutadditional software installations.

JavaScriptcanbewritten:

* InlinewithinHTMLtags(notrecommendedforlargeprojects),
* Within<script>tagsinsideHTMLfiles,
* Or,preferably,inseparate.jsfileslinkedtoHTMLdocuments.

### KeyConceptsinJavaScript

1. **VariablesandDataTypes**

JavaScript supports dynamic typing. Variables can hold various data types like strings, numbers, booleans,arrays,andobjects. ModernJavaScriptuseslet andconstinsteadof theoldervar keyword.

### Functions

Functionsencapsulatereusablecodeblocks.JavaScriptsupportsbothtraditionalandarrowfunction syntax. Functions are crucial for organizing code and executing actions when events occur.

### EventsandEventHandling

Eventsaretriggeredbyuseractions—likeclickingabuttonortypingintoaform.JavaScriptcanbeused to respond to these events, making the page more interactive and responsive.

### DOMManipulation

The**DocumentObjectModel(DOM)**isthebrowser'srepresentationofthepagestructure.JavaScript allows direct access to modify HTML elements, styles, or attributes dynamically based on logic or interaction.

### ControlStructures

JavaScriptincludesstandardcontrolflowstructureslikeif-else,switch,for,while,anddo-whileloops. These are essential for building logic-based interactions and responses in applications.

### JavaScriptandUserExperience

JavaScriptenhances**userexperience(UX)**by:

* + Reducingwaittimethroughliveupdatesanddynamicloading
  + Makingformsmoreuser-friendlywith instant validation
  + Allowingreal-timefeedback(likealertsorpop-ups)
  + Improvingnavigationand accessibility

Inthisproject,category-specificblogfilteringandinteractivebehaviorwereimplementedusing JavaScript, giving users a smooth and responsive interface.

### Conclusion

JavaScriptisapowerfulandessentialtoolformodernwebdevelopment.IttransformsstaticHTML pages into dynamic, interactive applications. Understanding JavaScript is critical for any developer aiming to build responsive and engaging web platforms.

BylearningJavaScript,developersgaintheabilityto:

* + Controlpagebehaviour
  + Enhanceuserinteraction
  + ConnectwithAPIs andbackends
  + Developfullyfunctionalwebapplications

JavaScriptformsthebridgebetweenwhatusersseeandhowtheyinteractwithyour application

### IntroductiontoNode.js

**WhatisNode.js?**

**Node.js**isanopen-source,cross-platformJavaScriptruntimeenvironmentthatallowsdeveloperstorun JavaScript code **outside the browser**, primarily on the **server-side**. Built on the **V8 JavaScript engine** used by Google Chrome, Node.js is known for its **speed**, **scalability**, and **non-blocking I/O model**, making it ideal for building **real-time, data-intensive web applications**.

### WhyUseNode.js?

* + **JavaScriptEverywhere**:Samelanguageforbothfrontendandbackend
  + **AsynchronousProgramming**:Handlesmultiplerequestswithoutwaiting
  + **LightweightandFast**:Optimizedforperformance
  + **ScalableArchitecture**:Supportsmicroservicesandreal-time systems
  + **RichEcosystem**:Poweredbynpm(NodePackageManager),whichoffersthousandsofreusable packages

### KeyFeaturesofNode.js

* + **Non-blockingI/O**:Executestasksinthebackgroundwithoutwaiting
  + **Event-drivenarchitecture**:Triggersactionsinresponsetoevents(e.g.,HTTPrequests)
  + **Single-threadedbutefficient**:Usesasingle-threadmodelbuthandlesconcurrencythrough events and callbacks
  + **Built-inmodules**:Providescorelibrarieslikehttp,fs,path,andosforbuildingapplications without external tools

### IntroductiontoExpress.js

**WhatisExpress.js?**

**Express.js**isaminimalistandflexiblewebapplicationframeworkforNode.js.Itsimplifiestheprocess of building server-side applications by providing a structured way to handle routes, middleware, and requests. Express is often referred to as the **de facto standard** framework for Node.js due to its simplicity and powerful features.

### WhyUse Express.js?

* + **SimplifiedRouting**:Easytodefineandmanageapplicationendpoints(e.g.,/login,/blog)
  + **MiddlewareSupport**:Easilyinsertlogicbetweenrequestandresponse(e.g.,authentication, logging)
  + **FastDevelopment**:Reducesboilerplatecodeand setuptime
  + **RESTfulAPISupport**:PerfectforbuildingAPIs tointeractwithfrontend apps
  + **Integration**:Workssmoothlywithdatabaseslike MongoDBusinglibrariessuchas Mongoose

### CoreConceptsinExpress.js

* + **Routes**:DefinehowtheserverrespondstodifferentHTTPrequests(GET,POST,PUT, DELETE).
  + **Middleware**:Functionsthatrunbeforeorafterroutehandlers;usedfortaskslikeauthentication or error handling.
  + **RequestandResponseObjects**:Expressprovidesreqandrestoaccessclientdataandsend responses.
  + **ViewEngine(Optional)**:CanrenderdynamicpagesusingtemplateengineslikeEJS,Pug,etc.

**“WANDER&WELLNESS:ACategory-BasedBlogPlatformwithUser-**

**CentricContent Management”**

**ProjectDescription:**

The **Wander & Wellness** blog platform is a full-stack web application developed to encouragecontentsharingandexplorationacrosswellness-focusedtopics.Theprimarygoal of the project is to provide users with an engaging space to read and publish blog articles under predefined lifestyle categories such as **Travel**, **Food**, **Adventure**, **Minimalism**, and **Self-Care**.

Theplatformallowsvisitorstoexploreacuratedcollectionof**defaultblogposts**organized category-wise.Registereduserscancontributetheirownblogpostsbyselectingacategory, enteringcontent,andoptionallyuploadinganimage.Eachuserhastheabilityto **view,edit, and delete** their own posts securely. This ensures that users have full control over their contributions, while maintaining a shared, community-driven experience for readers.

Theproject aimstocombine**contentdiscoveryandpersonalpublishing**,offeringauser- friendlyandsecureenvironmentthatpromoteswellnesseducationandcreativeexpression.

### OverviewoftheProject

Wander & Wellness is built using the **Node.js runtime** and the **Express.js framework** to handlebackendoperations.Itconnectstoa**MongoDB**databasethrough**Mongoose**,whichis used to manage blog posts and user credentials. The **frontend** is designed with HTML and CSS, and enhanced by category-specific JavaScript files that dynamically load and display blog content.

Theapplicationincludesthefollowingcomponents:

* + **UserAuthentication**:Session-basedloginandsignup system.
  + **PostManagement**:Authenticateduserscancreate,view,edit,anddeletetheirown posts.
  + **ContentCategorization**:Postsaresortedanddisplayedbycategories,improving readability and navigation.
  + **Dashboard**:Apersonalizedsectionwhereuserscanviewandmanagetheirrecent posts.
  + **ImageUploads**:HandledviatheMulterlibrary,allowinguserstoincludeimagesin their blog posts.

DataflowintheapplicationstartsfromthefrontendUIwhereusersinteractwithformsand buttons. Requests are handled by Express routes, which validate data, interact with the database, and return responses that are dynamically rendered or handled via frontend JavaScript.

Inshort,theprojectusesa**modularandscalablearchitecture**,makingiteasytomaintain, extend, and deploy for real-world use.

### ProjectRequirements

Thissectionoutlinesthesoftwareandhardwareprerequisitesnecessarytodevelop,run,and test the *Wander & Wellness* blog platform.

### SoftwareRequirements

* + **OperatingSystem:**Windows10/Linux/macOS
  + **Node.js:**v14 orabove–to runthebackend server
  + **MongoDB:**CommunityEdition–fordatabasemanagement
  + **CodeEditor:**VisualStudio Code– forwriting anddebugging code
  + **Browser:** GoogleChrome/Firefox– forfrontendtesting
  + **Postman:**(Optional)fortestingAPIroutesduringdevelopment

### HardwareRequirements

* + **Processor:**Inteli3or above
  + **RAM:**Minimum4 GB
  + **Storage:**Atleast500MBfreeforprojectfiles,database,anddependencies
  + **InternetConnection:**Requiredforinstallingnpmpackagesandtestingcloud deployment (if applicable)

### TechnologyStackUsed

The **Wander & Wellness** blog platform is developed using a **full-stack JavaScript architecture**,combiningfrontend,backend,anddatabasetechnologiestodeliveradynamic and user-focused blogging experience.

### Frontend(Client-Side)

##### HTML5

Usedtostructurethewebpagesincludingforms,blogcontentsections,and navigation elements.

##### CSS3

Appliedforstylingandlayoutdesigntomaketheinterfacevisuallyappealingand responsive.

##### JavaScript

Usedtohandleinteractivity,such as:

* + - Fetchinganddisplaying category-wiseblogposts dynamically.
    - UpdatingtheUIbasedonuseractions(e.g.,showing/hidingmessages, rendering posts).

### Backend(Server-Side)

##### Node.js

ProvidesaJavaScriptruntimeenvironmentforrunningserver-sidelogic.

##### Express.js

Aminimalist webframeworkusedto:

* + - DefineRESTful routes
    - HandleHTTPrequests(GET,POST,PUT,DELETE)
    - Managesessionsand middleware

##### Multer

Amiddlewarelibrary usedto handleimageuploadsassociated withblog posts.

### Database

##### MongoDB

ANoSQLdocument-orienteddatabaseusedtostore:

* + - Usercredentials
    - Blogposts(title,content,author,category,image paths)
    - Metadataliketimestamps

##### Mongoose

AnODM(ObjectDataModeling)libraryforMongoDBthatsimplifiesdatamodeling and schema validation.

### OtherToolsandLibraries

* + **Express-Session**:Formanaginguserloginsessionssecurely.
  + **FileSystem Module(fs)**:Usedtodeleteimagefileswhenblogpostsare removed.
  + **NodePackageManager (npm)**:Forinstallingandmanagingbackend dependencies.

### ModuleBreakdown

The **Wander & Wellness** blog platform is divided into multiple functional modules, each responsibleforhandlingaspecificaspectoftheapplication.Thismodularstructureensures clean code organization, ease of maintenance, and scalability.

### UserAuthenticationModule

#### Purpose:

Handlesusersign-up,login,sessionmanagement, andlogoutoperations.

#### KeyFeatures:

* + Userregistrationwithuniqueusernames.
  + Secureloginusingsession-basedauthentication.
  + Sessionpersistenceacrossuseractivities.
  + Logoutfeaturetoendthe session securely.

#### TechnologiesUsed:

* + Expresssessionmiddleware
  + MongoDBforstoringuser data
  + Mongooseusermodel

### BlogPostManagementModule

#### Purpose:

Allowsuserstocreate,view,edit,anddeletetheir ownblogposts.

#### KeyFeatures:

* + Blogcreationwithtitle,content,category, andoptionalimage.
  + Viewingallblogs,filteredbycategories.
  + Editinganddeleting onlypostsauthoredby thelogged-in user.
  + Imagehandlingviafileupload system.

#### TechnologiesUsed:

* + MongoDB forstoring blog entries
  + Multerforfile uploads
  + Expressroutesandcontrollers
  + Mongoose blog schema

### Category-BasedDisplayModule

#### Purpose:

Organizesblogcontentundermeaningfullifestyle categories.

#### KeyFeatures:

* + Category-specificJavaScriptfiles(travel.js,food.js,etc.)
  + Fetchesanddisplayspostsbasedonselected category
  + Enhancesuserexperiencebygroupingsimilarcontent

#### TechnologiesUsed:

* + FrontendJavaScript
  + RESTAPIendpoints
  + DynamicDOMmanipulation

### DashboardandProfileModule

#### Purpose:

Providesuserswithapersonalizedviewoftheiraccountandrecentactivity.

#### KeyFeatures:

* + Displaystotal numberofpostswrittenby theuser.
  + Showsrecentlypublished posts.
  + Providesaccesstoedit/deleteoptionsforeachpost.
  + Personalizesthedashboardwithusername andcontenthistory.

#### TechnologiesUsed:

* + BackendAPIendpoint(/profile-data)
  + Frontenddynamic rendering
  + MongoDBqueriesforcountingandlisting posts

### FileUploadandMediaManagementModule

#### Purpose:

Handlesimageuploadsforblogpostsandmanagesfile storage.

#### KeyFeatures:

* + Allowsuserstouploadanimagewiththeirblog.
  + Savesimagesontheserver inadedicateduploadsfolder.
  + Deletesimage filesautomaticallywhentheassociatedpostisdeleted.

#### TechnologiesUsed:

* + Multer(Node.jsmiddleware)
  + Filesystem(fsmoduleinNode.js)
  + Expressrouteintegration

### FrontendUIandNavigationModule

#### Purpose:

Providesaresponsive anduser-friendlyinterface forinteraction.

#### KeyFeatures:

* + HTMLandCSS-based layoutforpageslikelogin,dashboard, categories
  + JavaScript-enhancednavigationandinteractivity
  + StaticassetmanagementviaExpress

#### TechnologiesUsed:

* + HTML5,CSS3
  + JavaScript(DOM,events)
  + Expressstaticfile serving

### SystemArchitectureofWander&WellnessBlogPlatform

ThesystemarchitectureoftheWander&Wellnessblogplatformfollowsaclassicthree-tier architecture, consisting of:

1. PresentationLayer(Frontend)
2. ApplicationLayer(Backend)
3. DataLayer(Database)

Thisstructureensuresaclearseparationofresponsibilitiesandenhancesmaintainability, scalability, and security.

### PresentationLayer(Frontend)

Thisistheuserinterface thatusersinteractwiththroughawebbrowser.

##### Technologies:

* + HTML5– Forstructuring web content(forms, buttons, pages)
  + CSS3–For stylingandlayout ofUIcomponents
  + JavaScript–Forinteractivity(fetching,displaying,updatingblogposts)

##### Responsibilities:

* + Displaylogin/signuppages,blogfeeds,dashboards
  + Captureuserinput (writingposts,selecting categories)
  + CommunicatewithbackendusingHTTPrequests
  + DynamicallyupdatethepageusingDOM manipulation

### ApplicationLayer(Backend)

Thislayercontainsthebusinesslogicandserver-side operations.

##### Technologies:

* + Node.js–Runtimeenvironmenttoexecuteserver-sidecode
  + Express.js–Frameworktohandleroutes,middleware,and APIs
  + Multer–For handlingimageuploads
  + Express-Session–Tomanageuserloginsessions

##### Responsibilities:

* + Handlerouting(login,signup,create/edit/delete/view blogs)
  + Authenticateand authorize users
  + Processimageuploads
  + Manageinteractionsbetweenfrontendand database

### DataLayer(Database)

Thislayerstoresallpersistentdatafortheapplication. Technologies:

* + MongoDB– NoSQLdatabasetostore documents
  + Mongoose– ODMforschema modelingand databaseoperations

##### Collections:

* + users:Storesusercredentials(username, password)
  + blogs:Storesblogcontent(title,body,author,category,date,imagepath)

##### Responsibilities:

* + Saveandretrieveblogpostsanduser data
  + Enforceschemavalidation
  + Enablefastqueryingbasedonauthororcategory

### DataFlowOverview:

1. Useropensthe website(frontendloads).
2. UserinteractswiththeUI(e.g.,logsinorcreates a blog).
3. Frontendsends HTTPrequeststothebackendvia routes.
4. Backendprocessesthe request,communicateswithMongoDBifneeded.
5. Data(e.g.,bloglist,confirmation)isreturnedtothefrontend.
6. JavaScriptupdatestheDOM(DocumentObjectModel)toreflectthe change.

### KeyFeatures

The Wander and Wellness blog platform was developed to provide users with a seamless experienceforsharingandmanagingwellness-relatedcontent.Belowisadetaileddescription of the key features implemented:

#### UserAuthenticationSystem

Asecureandreliableauthenticationsystemallowsuserstoregisterandlogintotheplatform. Sessionsaremanagedusingcookies,ensuringthatonlyauthenticateduserscanaccesscertain routes like the dashboard, post creation, and editing functionalities.

#### CreateandManageBlogPosts

Userscancreateblogpostsbyenteringatitle,content,anduploadinganimage.Thepostis savedinMongoDBandrendereddynamicallyacrossthedashboardandprofilepages.This feature supports expressive storytelling with both text and visual content.

#### ImageUploadCapability

Eachblogpostcanincludeanimageuploadedbytheuser.Theimageisstoredontheserver using Multer, a Node.js middleware for handling multipart/form-data. Uploaded images are rendered in cards beside the post content, enhancing user engagement.

#### ResponsiveUserDashboard

Thedashboardinterfaceprovidesaclear,visuallymodernlayoutforuserstoview,create, and manage their posts. It is styled using Bootstrap 5 for responsiveness across devices, making it accessible on desktops, tablets, and smartphones.

#### EditandDeleteFunctionality

Userscaneditordeleteonlythepoststheyhavecreated.Thisisenforcedthroughbackend checks using user authentication. Posts can be updated with new content or images and removed completely when no longer relevant.

#### ProfileView

Each user has access to a personalized profile page that displays only their blog posts. This pagehelpsuserskeeptrackoftheirpublishedcontentandactsas adigitalportfoliowithinthe wellness community.

#### MongoDBandMongooseIntegration

AlluserandblogdataarestoredinMongoDBusingMongooseastheObjectDataModeling (ODM) library. This provides a flexible schema design and efficient interaction with the database for CRUD operations.

#### Bootstrap-basedUI

The entire platform uses Bootstrap 5 for a consistent, clean, and mobile-first UI design. Componentssuchascards,forms,modals,andnavigationbarsareusedtomaketheinterface intuitive and easy to use.

#### ModernProjectArchitecture

The project uses a modular folder structure with separation of concerns between routes, models,publicassets,andviews.Thisimprovesmaintainabilityandscalabilityforfuture development.

### FutureEnhancementsoftheProject

ToevolveWander&Wellnessintoamorecomprehensiveplatform,thefollowingfeatures can be considered for future development:

#### TechnicalImprovements

* + **JWT-basedAuthentication:**Replacesession-basedloginwithtoken-based authentication for better scalability.
  + **CloudDatabaseandFileHosting:**MigrateMongoDBandfileuploadstocloud platforms (MongoDB Atlas, Cloudinary, Firebase Storage).

#### FeatureEnhancements

* + **RichTextEditor:**Replaceplaintextinputswitheditorsthatsupportformatting, links, media, and markdown.
  + **UserProfiles:**Enablepublicprofilepageswherevisitorscanviewpostsbyaspecific author.
  + **Follow/SubscribeFeature:**Letusersfollowotherwritersorcategoriesandget

notifiedofnew content.

* + **CommentandLikeSystem:**Addinteractivitybyallowinguserstoengagewith posts.
  + **AdvancedSearchandFilters:**Addsearchfunctionalitywithfiltersforauthor,date, or keywords.

### Code:

**server.js**

constexpress= require("express");

const session = require("express-session"); constbodyParser=require("body-parser"); const mongoose = require("mongoose"); const multer = require("multer");

constpath=require("path"); const fs = require("fs");

constUser=require("./models/User"); constBlog = require("./models/Blog");

constapp=express(); const PORT = 3000;

//ConnecttoMongoDBAtlas mongoose

.connect("mongodb+srv://Charitha:charitha\_7777@cluster0.rqpk81d.mongodb.net/mybloga pp?retryWrites=true&w=majority&appName=Cluster0", {

useNewUrlParser: true, useUnifiedTopology:true,

})

.then(()=>console.log("ConnectedtoMongoDBAtlas"))

.catch((err)=>console.error("MongoDBconnectionerror:",err));

//Middleware

app.use(bodyParser.urlencoded({extended:true})); app.use(bodyParser.json());

app.use(session({secret:"secret",resave:false,saveUninitialized:true})); app.use(express.static("public"));

app.use("/uploads",express.static(path.join(dirname,"uploads")));//forservingimages

//Setup multer for file uploadsconststorage=multer.diskStorage({ destination: (req, file, cb) => {

constuploadPath =path.join(dirname,"uploads");

if(!fs.existsSync(uploadPath))fs.mkdirSync(uploadPath); cb(null, uploadPath);

},

filename:(req,file,cb) =>{

constuniqueName=`${Date.now()}-${file.originalname}`; cb(null, uniqueName);

},

});

constupload=multer({storage});

//Dashboard Access app.get("/dashboard.html",(req,res,next)=>{

if(!req.session.username)returnres.redirect("/login.html"); next();

});

//Sign up

app.post("/signup", async (req, res) => { const{username,password}=req.body;

constuserExists=awaitUser.findOne({username});

if(userExists)returnres.send("Useralreadyexists.<ahref='/signup.html'>Tryagain</a>"); await User.create({ username, password });

res.redirect("/login.html");

});

//Login

app.post("/login", async (req, res) => { const{username,password}=req.body;

constuser=awaitUser.findOne({username}); if (!user || user.password !== password)

returnres.send("Invalidcredentials.<ahref='/login.html'>Tryagain</a>"); req.session.username = username;

res.redirect("/dashboard.html");

});

//Logout

app.get("/logout", (req, res) => { req.session.destroy(()=>res.redirect("/login.html"));

});

//CreatePost (withimage upload)

app.post("/post",upload.single("image"),async(req,res)=>{ const username = req.session.username;

if(!username)returnres.status(401).send("Login required");

const{ title,content }=req.body;

constimage=req.file? `/uploads/${req.file.filename}`: null;

awaitBlog.create({title,content,image,author:username,date:newDate()}); res.redirect("/dashboard.html");

});

//FetchAllBlogs

app.get("/blogs",async(req,res)=>{

constblogs=awaitBlog.find().sort({date: -1});

res.json({blogs,currentUser:req.session.username||null});

});

//Edit Post

app.post("/edit-post",async(req,res)=>{ const { id, title, content } = req.body; const blog = await Blog.findById(id);

if(!blog||blog.author!==req.session.username) return res.status(403).send("Unauthorized");

blog.title = title; blog.content = content; blog.date=newDate(); await blog.save(); res.sendStatus(200);

});

//DeletePost

app.post("/delete-post",async(req,res)=>{ const { id } = req.body;

constblog=awaitBlog.findById(id);

if(!blog||blog.author!==req.session.username) return res.status(403).send("Unauthorized");

//Alsoremoveuploadedimagefileifexists if (blog.image) {

const imgPath = path.join(dirname, blog.image); if(fs.existsSync(imgPath))fs.unlinkSync(imgPath);

}

awaitBlog.findByIdAndDelete(id); res.sendStatus(200);

});

//ProfileData

app.get("/profile-data",async(req,res)=>{ const username = req.session.username;

if(!username)returnres.status(401).send("Unauthorized");

consttotalPosts=awaitBlog.countDocuments({author:username});

constrecentPosts=awaitBlog.find({author:username}).sort({ date: -1}).limit(5);

res.json({username,totalPosts,recentPosts });

});

//Start Server

app.listen(PORT,()=>console.log(`7'•˙¸.sServerrunningat[http://localhost:${P](http://localhost/)ORT}`));

### dashboard.html

<!DOCTYPEhtml>

<htmllang="en">

<head>

<metacharset="UTF-8"/>

<metaname="viewport"content="width=device-width,initial-scale=1.0"/>

<title>UserDashboard</title>

<style>

/\*Basic Resetand BoxSizing \*/

\* {

box-sizing:border-box; margin: 0;

padding:0;

}

/\*GeneralBodyStyling\*/ body {

font-family:'SegoeUI',sans-serif; background-color: #f5f5f5;

}

/\*PageContainer \*/

.container { width:100%;

padding:1.5rem2vw;

}

/\*TopNavigation Bar Styling \*/

.top-bar{

background-color:#343a40; color: white;

padding:1rem2vw; display: flex;

justify-content:space-between; align-items: center;

}

.top-barh2{ margin: 0;

}

/\*NavigationMenu \*/

.navbar-nav { list-style:none; display: flex; gap: 1.5rem;

}

.navbar-nava{

color:white;

text-decoration:none; font-weight: bold;

}

/\*FormStylingforBlogSubmission\*/ form {

background:white; padding: 2rem; margin: 2rem 0; border-radius:8px;

box-shadow:0 0 10px rgba(0,0,0,0.05);

width: 100%;

}

formh3{

margin-bottom:1rem;

}

forminput,formtextarea{ width: 100%;

padding: 0.75rem; margin-bottom: 1rem; border:1pxsolid#ccc; border-radius: 6px; font-size: 1rem;

}

forminput[type="file"]{ padding: 0.3rem;

}

formbutton{

padding: 0.6rem 1.5rem; background-color:#007bff; color: white;

border: none; border-radius:4px; font-size: 1rem; cursor: pointer;

}

/\*BlogGridDisplay\*/ #blogs {

display: grid;

grid-template-columns:repeat(auto-fit,minmax(350px,1fr)); gap: 1.5rem;

width: 100%;

}

/\*Blog PostCard Styling \*/

.post {

background: white; border-radius:10px;

box-shadow:05px15pxrgba(0,0,0,0.1); overflow: hidden;

transition:transform 0.2s;

}

.post:hover{

transform:translateY(-5px);

}

.post img {width: 100%; height: 220px; object-fit:cover;

}

/\*BlogPostContentSection \*/

.post-content { padding:1rem;

}

.post-contenth4{ color: #9c27b0;

margin-bottom:0.5rem;

}

.post-contentp{ color: #555;

}

.post-contentsmall{ color: #888; display: block; margin: 0.5rem 0;

}

/\*ButtonsforBlogActions\*/

.btn-edit,

.btn-delete,

.btn-read{

padding:0.5rem1rem; font-size: 0.85rem; border: none;

border-radius: 4px; margin-right:0.5rem; cursor: pointer;

}

.btn-edit{

background-color:#ffc107; color: white;

}

.btn-delete{

background-color:#dc3545; color: white;

}

.btn-read{

background-color:#9c27b0; color: white;

}

/\*ResponsiveDesignforSmallScreens\*/ @media (max-width: 600px) {

.top-bar{

flex-direction:column; align-items: flex-start; gap: 1rem;

}

.navbar-nav{

flex-direction:column;

}

}

</style>

</head>

<body>

<!--Top NavigationBar-->

<divclass="top-bar">

<h2>Dashboard</h2>

<nav>

<ulclass="navbar-nav">

<li><ahref="index.html">Home</a></li>

<li><ahref="profile.html">Profile</a></li>

<li><ahref="/logout">Logout</a></li>

</ul>

</nav>

</div>

<divclass="container">

<!--BlogPostSubmissionForm -->

<formaction="/post"method="POST"enctype="multipart/form-data">

<h3>PostaNew Blog</h3>

<inputtype="text"name="title"placeholder="Title"required/>

<textareaname="content"rows="4"placeholder="Content..."required></textarea>

<inputtype="file"name="image"accept="image/\*"/>

<buttontype="submit">Publish</button>

</form>

<!--SectionforDisplayingAllBlogs-->

<section>

<h3style="margin-bottom:1rem;">All Blogs</h3>

<divid="blogs"></div><!--Blogswillbeloadedheredynamically -->

</section>

</div>

<!--ScriptforFetchingandManagingBlogPosts-->

<script>

fetch("/blogs")//Requesttogetallblogs

.then(res=>res.json())

.then(({blogs,currentUser})=>{

constcontainer=document.getElementById("blogs");

blogs.forEach(blog=>{

constdiv=document.createElement("div"); div.className = "post";

constisOwner= currentUser===blog.author; //Check ifcurrentuseristheauthor

//CreateblogcardHTMLdynamically div.innerHTML = `

${blog.image?`<imgsrc="${blog.image}"alt="Blogimage"/>`: ''}

<divclass="post-content">

<h4>${blog.title}</h4>

<p>${blog.content}</p>

<small>by${blog.author}–${newDate(blog.date).toLocaleDateString()}</small>

${isOwner?`

<buttonclass="btn-edit"data-id="${blog.\_id}"data-title="${blog.title}"data- content="${blog.content}">Edit</button>

<buttonclass="btn-delete"data-id="${blog.\_id}">Delete</button>

` : `

<buttonclass="btn-read">Read More</button>

`}

</div>

`;

container.appendChild(div);// Addblog cardto the container

});

//EventdelegationforEditandDeletebuttons container.addEventListener("click", e => {

if(e.target.matches(".btn-delete")){ fetch("/delete-post", {

method: "POST",

headers:{"Content-Type":"application/json"}, body: JSON.stringify({ id: e.target.dataset.id })

}).then(()=>location.reload());//Reloadtoreflectdeletion

}

if(e.target.matches(".btn-edit")){ const id = e.target.dataset.id;

consttitle=prompt("Newtitle:", e.target.dataset.title);

constcontent=prompt("Newcontent:",e.target.dataset.content); if (title && content) {

fetch("/edit-post",{ method: "POST",

headers:{"Content-Type":"application/json"}, body: JSON.stringify({ id, title, content })

}).then(()=>location.reload());//Reloadtoreflectupdate

}

}

});

});

</script>

</body>

</html>

### signup.html

<!DOCTYPEhtml>

<htmllang="en">

<head>

<metacharset="UTF-8">

<metaname="viewport"content="width=device-width,initial-scale=1.0">

<title>Signup</title>

<linkrel="stylesheet"href="styles.css">

<style>body {

font-family:Arial,sans-serif; background-color: #f8f9fa; margin: 0;

padding: 0; display:flex;

align-items: center; justify-content:center; height: 100vh;

}

a{

color:#007bff;

text-decoration: none;

}

a:hover{

text-decoration:underline;

}

.sign-form h2 { text-align:center;

margin-bottom:20px;

}

.sign-form input { width: 100%; padding: 0.75rem; margin-bottom:1rem;

border:1pxsolid#ccc; border-radius: 4px;

}

.sign-formbutton{ width: 100%; padding:0.75rem;

background-color:#007bff; color: white;

border: none; border-radius:4px; cursor: pointer; font-size: 1rem;

}

.sign-form button:hover { background-color:#0056b3;

}

.sign-formp{

text-align: center; margin-top:1rem;

}

</style>

</head>

<body>

<divclass="sign-form">

<h2>CreateYourAccount</h2>

<formaction="/signup"method="POST">

<inputtype="text"name="username"placeholder="Username"required>

<inputtype="password"name="password"placeholder="Password"required>

<buttontype="submit">Sign Up</button>

</form>

<p>Alreadyhaveanaccount?<ahref="login.html">Login</a></p>

</div>

</body>

</html>

### login.html

<!DOCTYPEhtml>

<htmllang="en">

<head>

<metacharset="UTF-8"/>

<metaname="viewport"content="width=device-width,initial-scale=1.0"/>

<title>Login</title>

<linkrel="stylesheet"href="styles.css"/>

<style>body {

font-family:Arial,sans-serif; background-color: #f8f9fa; margin: 0;

padding: 0; display:flex;

align-items: center; justify-content:center; height: 100vh;

}

a{

color:#007bff;

text-decoration: none;

}

a:hover{

text-decoration:underline;

}

.sign-form { background:white; padding: 2rem; border-radius:8px;

box-shadow:0 0 10px rgba(0,0,0,0.1);

width: 100%;

max-width: 400px;

}

.sign-form h2 { text-align:center;

margin-bottom:20px;

}

.sign-form input { width: 100%; padding: 0.75rem; margin-bottom:1rem;

border:1pxsolid#ccc; border-radius: 4px;

}

.sign-formbutton{ width: 100%; padding:0.75rem;

background-color:#007bff; color: white;

border: none; border-radius:4px; cursor: pointer; font-size: 1rem;

}

.sign-form button:hover { background-color:#0056b3;

}

.sign-formp{

text-align: center; margin-top:1rem;

}

</style>

</head>

<body>

<divclass="sign-form">

<h2>LogintoYour Account</h2>

<formaction="/login"method="POST">

<inputtype="text"name="username"placeholder="Username"required/>

<inputtype="password"name="password"placeholder="Password"required/>

<buttontype="submit">Login</button>

</form>

<p>Don'thavean account? <ahref="signup.html">Sign up</a></p>

</div>

</body>

</html>

### index.html

<!DOCTYPEhtml>

<htmllang="en">

<head>

<metacharset="UTF-8"/>

<metaname="viewport"content="width=device-width,initial-scale=1.0">

<title>Wander& Wellness</title>

<linkhref=["https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"](https://cdn.jsdelivr.net/npm/bootstrap%405.3.0/dist/css/bootstrap.min.css) rel="stylesheet" />

<linkrel="stylesheet"href="/styles.css"/>

</head>

<body>

<!--TopBar-->

<divclass="top-barcontainerd-flexjustify-content-between align-items-centerpy-3">

<buttononclick="scrollToSubscribe()"class="btnbtn-primary">Subscribe</button>

<h1class="logom-0"><ahref="/index.html"style="text-decoration:none;color: inherit;">Wander & Wellness</a></h1>

<ahref="/signup.html"class="btnbtn-success">Sign up</a>

</div>

<!--Navigation-->

<navclass="navbarnavbar-expand-lgnavbar-lightbg-whiteborder-bottom">

<divclass="container">

<ulclass="navbar-nav mx-auto">

<liclass="nav-itempx-2"><ahref="/travel.html"class="nav-link">Travel</a></li>

<liclass="nav-itempx-2"><ahref="/food.html"class="nav-link">Food</a></li>

<liclass="nav-itempx-2"><ahref="/wellness.html"class="nav- link">Wellness</a></li>

<liclass="nav-itempx-2"><ahref="/adventure.html"class="nav- link">Adventure</a></li>

<liclass="nav-itempx-2"><ahref="/minimalism.html"class="nav- link">Minimalism</a></li>

<liclass="nav-itempx-2"><ahref="/selfcare.html"class="nav-link">Self- Care</a></li>

</ul>

</div>

</nav>

<!--MainContent-->

<mainclass="container mt-4">

<divid="wellness-tip-root"></div>

<!--Featured-->

<divclass="featured-containermb-4">

<sectionclass="featuredp-4bg-darktext-whiterounded">

<h2>TopTravelDestinationsfor2025</h2>

<p>Discoverbreathtaking,underratedtravel destinationsyouneedtovisit this year.</p>

<ahref="/travel.html"class="read-moretext-warning">Continuereading...</a>

<divclass="extra-infomt-2">Fromsecludedbeachestocharmingmountainvillages, explore destinations that are off the beaten path and perfect for adventure seekers.</div>

</section>

</div>

<!--Posts-->

<sectionclass="rowg-4">

<divclass="col-md-6d-flexalign-items-center">

<divclass="postp-3bg-whiteshadowroundedw-100">

<h3><spanclass="badgebg-primary">Travel</span>Budget-FriendlyGetaways</h3>

<p>Apr4</p>

<p>Learnhowtotraveltheworldwithoutbreakingthebank.</p>

<ahref="/travel.html"class="read-moretext-primary">Continuereading</a>

<divclass="extra-infomt-2">Gettipsonaffordabledestinations,budgetairlines,and

cost-savingtravelhacksforanunforgettableyeteconomicaltrip.</div>

</div>

<divclass="thumbnailms-3">

<imgsrc="/images/budget\_travel.jpg"alt="BudgetTravel"class="img-fluid rounded">

</div>

</div>

<divclass="col-md-6d-flexalign-items-center">

<divclass="postp-3bg-whiteshadowroundedw-100">

<h3><spanclass="badgebg-success">Wellness</span>MindfulTravel</h3>

<p>Apr3</p>

<p>Howtotravelstress-freeand embracea mindfuljourney.</p>

<ahref="/wellness.html"class="read-moretext-success">Continuereading</a>

<divclass="extra-infomt-2">Learnhowtoincorporatemindfulnessintoyourtravels, from digital detox tips to meditation-friendly destinations.</div>

</div>

<divclass="thumbnailms-3">

<imgsrc="/images/mindfulness.jpg"alt="MindfulTravel"class="img-fluidrounded">

</div>

</div>

</section>

<!--HorizontalScrollSection-->

<sectionclass="horizontal-scroll-sectionmt-5">

<h3class="section-heading">ExploreMore</h3>

<divclass="scroll-container">

<divclass="scroll-cardfood">

<h4>Food</h4>

<p>Explorenutritiousrecipesandmindfuleatingtips.</p>

<ahref="/food.html"class="btnbtn-outline-dark">ReadMore</a>

</div>

<divclass="scroll-cardadventure">

<h4>Adventure</h4>

<p>Thrillingjourneysandoutdoorescapadesawait you.</p>

<ahref="/adventure.html"class="btnbtn-outline-dark">Discover</a>

</div>

<divclass="scroll-cardminimalism">

<h4>Minimalism</h4>

<p>Learntolive morewithless — declutteryourlife.</p>

<ahref="/minimalism.html"class="btnbtn-outline-dark">Explore</a>

</div>

<divclass="scroll-cardselfcare">

<h4>Self-Care</h4>

<p>Relax,recharge,andbuildyourbestroutines.</p>

<ahref="/selfcare.html"class="btnbtn-outline-dark">Start Now</a>

</div>

</div>

</section>

</main>

<!--SubscribeSection -->

<sectionid="subscribe"class="subscribe-sectionpy-5bg-lighttext-centerborder-top border-warning mt-5">

<h2class="mb-3">SubscribetoWander&Wellness</h2>

<pclass="text-mutedmb-4">Getthelatesttraveltipsandwellnessstoriesinyour inbox!</p>

<divclass="d-flexjustify-content-centerflex-wrapgap-2">

<inputtype="email"id="emailInput"class="form-controlw-auto"placeholder="Enter your email" style="min-width: 250px;" />

<buttonid="subscribeBtn"class="btnbtn-warning">Subscribe</button>

</div>

<pid="subscribeMessage"class="mt-3fw-bold"></p>

</section>

<!--Footer-->

<footerclass="bg-whitetext-centerp-4mt-4">

<sectionclass="about">

<h4>About</h4>

<p>Wander&Wellnessisyourgo-tosource fortravelinspirationandself-care insights.

We bring you the best travel tips, destination guides, and wellness practices to help you exploretheworldwhiletakingcareofyourself.Whetheryouseekadventureorrelaxation, we have something for you. Happy travels!</p>

</section>

</footer>

<!--Scripts-->

<script src=["https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"><](https://cdn.jsdelivr.net/npm/bootstrap%405.3.0/dist/js/bootstrap.bundle.min.js)/script>

<script>

functionscrollToSubscribe(){

constsection=document.getElementById("subscribe"); section.scrollIntoView({ behavior: "smooth" });

}

document.addEventListener("DOMContentLoaded",()=>{

const subscribeBtn = document.getElementById("subscribeBtn"); const emailInput = document.getElementById("emailInput"); constmessage=document.getElementById("subscribeMessage");

functionisValidEmail(email){

constemailPattern=/^[^\s@]+@[^\s@]+\.[^\s@]+$/; return emailPattern.test(email);

}

subscribeBtn.addEventListener("click",()=>{

constemail=emailInput.value.trim(); if (email === "") {

message.textContent="Emailisrequired."; message.style.color = "red";

}elseif(!isValidEmail(email)){

message.textContent="Pleaseenteravalidemailaddress."; message.style.color = "red";

}else {

message.textContent="Thankyouforsubscribing!"; message.style.color = "green";

emailInput.value="";

}

});

});

</script>

<!--ReactWellnessTip-->

<scriptsrc=["https://unpkg.com/react@18/umd/react.development.js"](https://unpkg.com/react%4018/umd/react.development.js)crossorigin></script>

<scriptsrc=["https://unpkg.com/react-dom@18/umd/react-dom.development.js"](https://unpkg.com/react-dom%4018/umd/react-dom.development.js)crossorigin></script>

<script type="text/javascript"> conste=React.createElement;

functionWellnessTip(){ return e('section', { style: {

display: 'flex', alignItems:'center', maxWidth:'600px',

margin:'20pxauto40pxauto', padding: '12px 20px', backgroundColor: '#f0f4f8', borderRadius: '8px',

boxShadow:'01px4pxrgba(0,0,0, 0.05)',

color:'#555',

fontFamily:"'SegoeUI',Tahoma,Geneva,Verdana,sans-serif"

}

}, [

e('div',{ style: {

fontSize: '2rem', marginRight:'15px', opacity: 0.6

}

},'x¸·^±·˙’'ˇ˜⏷'),

e('div',null,[

e('h4',{

style: { margin:0,

fontWeight:600,

fontSize:'1.2rem', color: '#444'

}

},'Wellness Tip'),

e('p',{

style: { marginTop:'6px', fontSize: '1rem', lineHeight: 1.4,

color:'#666'

}

},'Takeamomentdailytobreathedeeplyandresetyourmind.Mindfulnessenhances every travel experience.')

])

]);

}

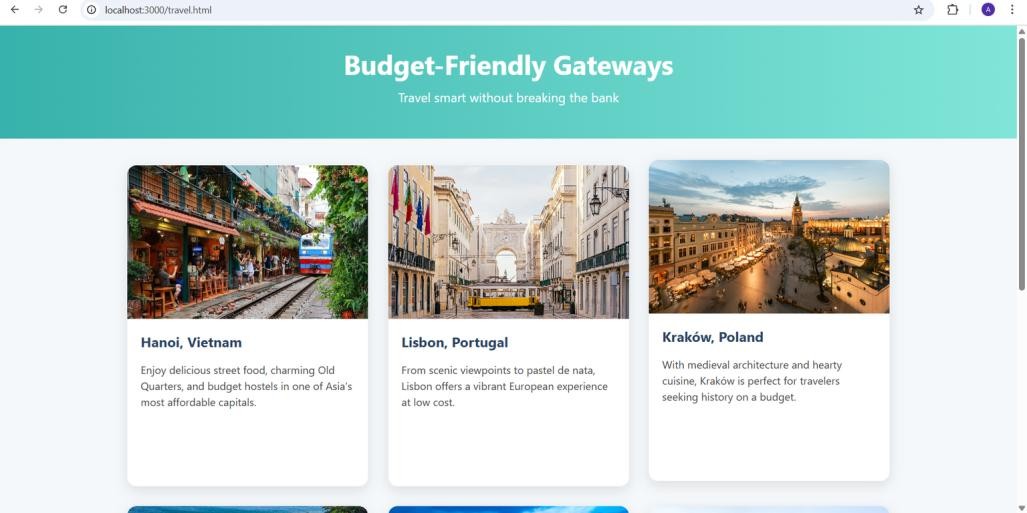
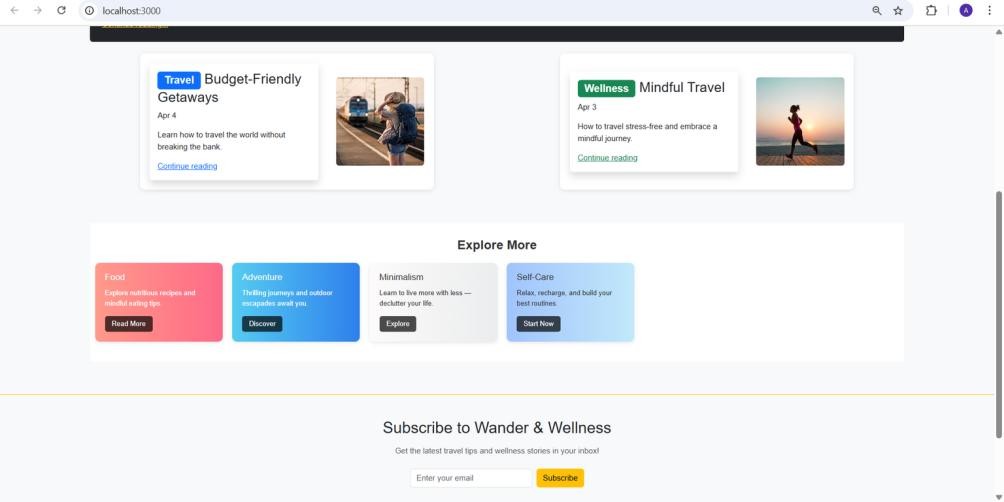
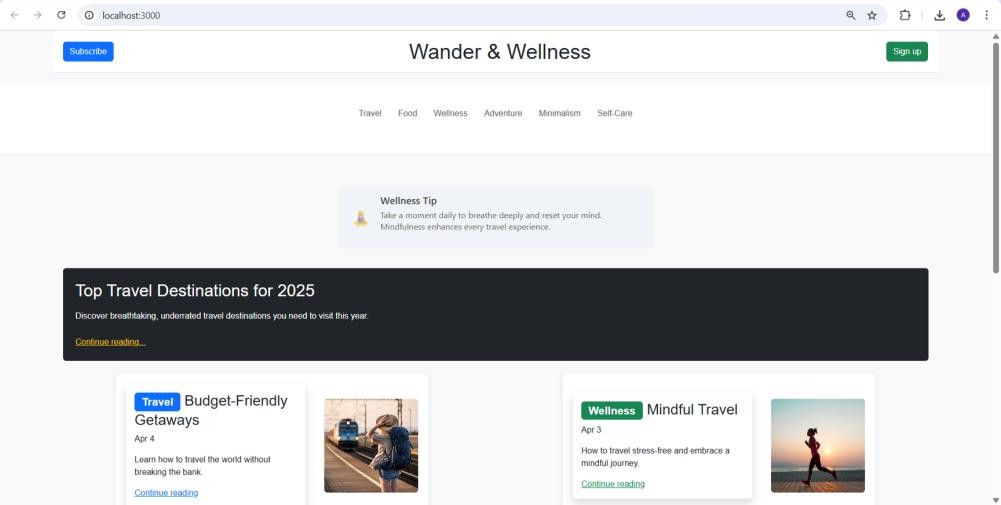
ReactDOM.createRoot(document.getElementById('wellness-tip- root')).render(e(WellnessTip));

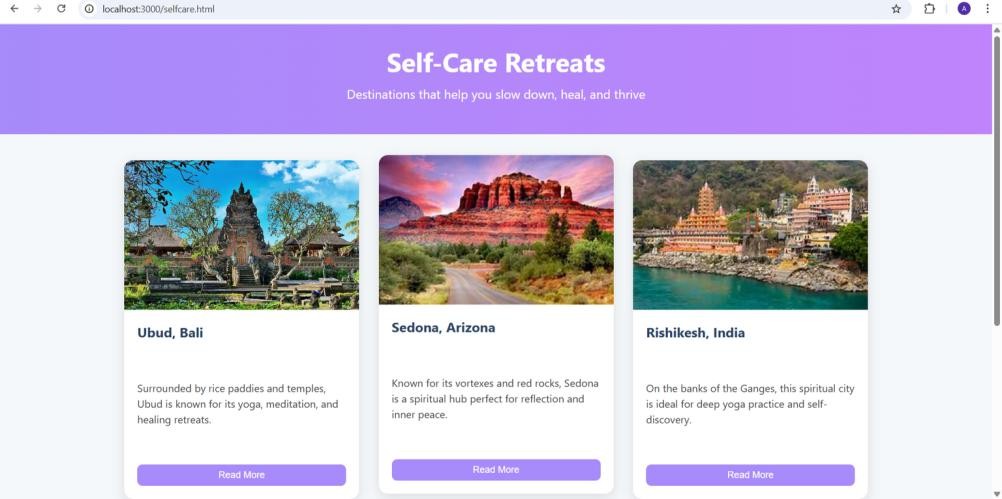
</script>

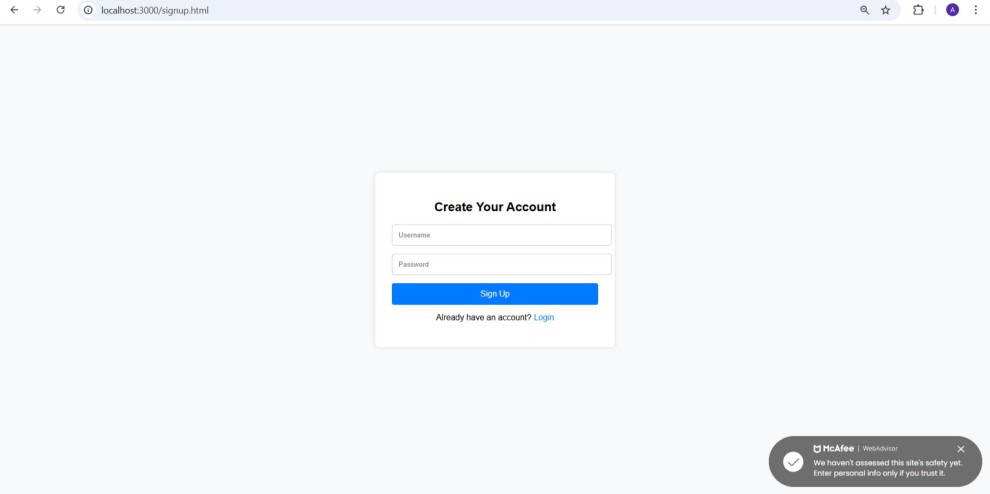
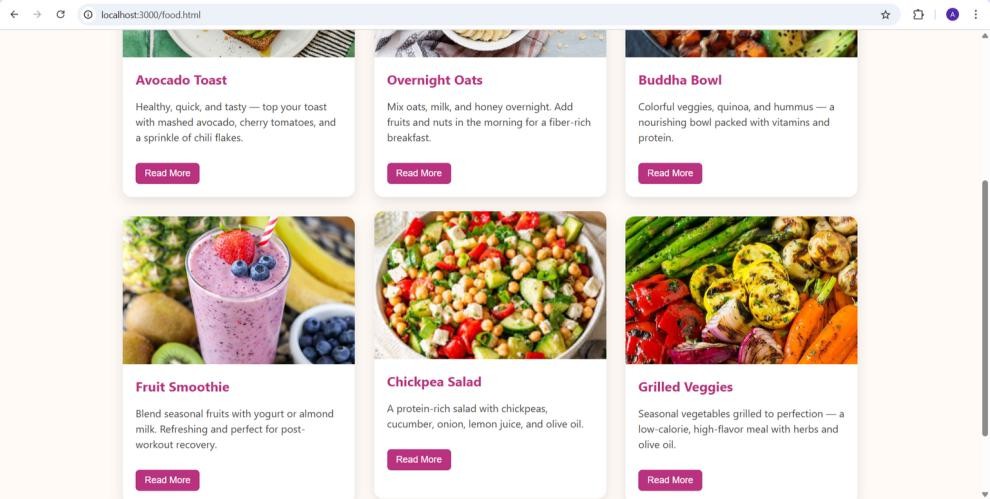
</body>

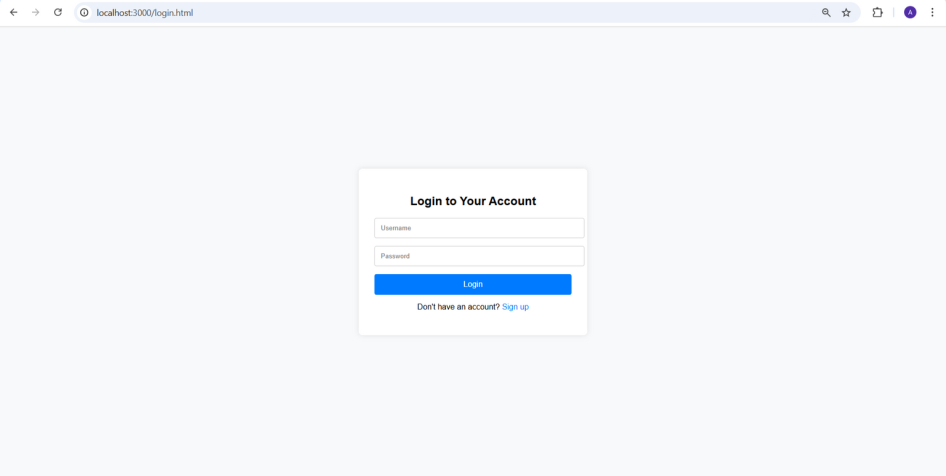
</html>

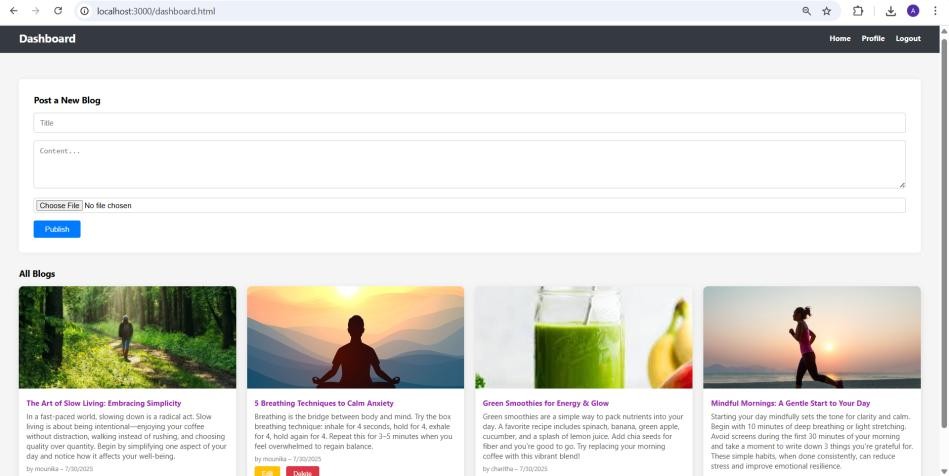
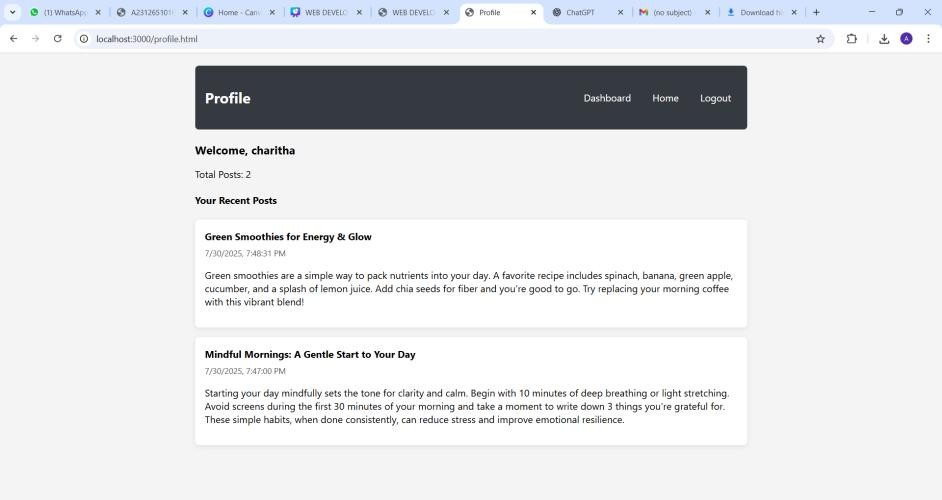
### Output:

****



****



****

**Advantages**

##### User-FriendlyInterface

* + Simple,cleandashboardmakesiteasyforuserstocreateandmanageblogposts without technical knowledge.

##### FullCRUD Functionality

* + UserscanCreate,Read,Update,andDeleteposts —givingthemfullcontrolover their content.

##### Category-BasedBrowsing

* + Blogsarecategorized,allowinguserstoexplorecontentbyinterest,improving content discoverability.

##### AuthenticationSupport

* + Onlyregistereduserscanpost,edit,ordelete,ensuringcontentsecurityand personalized dashboards.

##### ImageUpload Capability

* + Userscanenhancetheir blogsvisuallybyuploadingimages.

##### ResponsiveDesign

* + Thelayout adjustsfordifferent screen sizes,making it mobile-friendly.

##### Modular Architecture

* + TheuseofNode.js,Express.js,andMongoDBallowsforscalableandmaintainable code.

### Disadvantages

##### Basic Image Handling

* + Uploadedimagesarenotoptimizedorvalidateddeeply(e.g.,sizelimit,filetype filtering).

##### NoSearchorFilter

* + Userscan't**searchforspecificposts**orapply**advancedfilters**,limitingnavigation for larger content volumes.

##### NoCommentingorInteraction

* + There's**nosocialinteraction**(likecomments,likes,orshares)toboostuser engagement.

##### Session-BasedAuth(NoJWT)

* + Authenticationusesserver-sidesessionsinsteadofmorescalabletoken-based(JWT) systems.

##### NoAdminPanel

* + There'sno**admin role**formoderatingormanagingusers/content.

### Conclusion

The development of the **Wander & Wellness** blog platform has provided a comprehensive understanding of full-stack web development and practical implementation of real-world functionality. The project successfully achieves its aim of enabling users to explore a wide rangeofpreloadedwellness-relatedblogpostswhilealsoempoweringthemto**create,view, edit, and delete** their own content under various lifestyle categories.

By integrating technologies such as **Node.js**, **Express.js**, **MongoDB**, and **client-side JavaScript**,thesystemdemonstratesamodular,scalable,anduser-friendlysolution.Key features like **category-based filtering**, **user authentication**, **content management**, and **secure access control** contribute to a seamless blogging experience.

Throughoutthedevelopmentprocess,severalskillsweregained—includingfrontenddesign, backend logic implementation, RESTful API design, session handling, and database management. The system architecture follows a clear separation of concerns, ensuring that the application is maintainable and extendable for future enhancements.

Overall, this project not only meets its core objectives but also lays a solid foundation for buildingmoreadvancedcontent-sharingplatformsinthefuture.Itshowcaseshowwellness content can be organized, personalized, and securely managed in a dynamic web environment.

### References:

* + [**https://www.mongodb.com/docs/**](https://www.mongodb.com/docs/)
  + [**https://expressjs.com/**](https://expressjs.com/)
  + [**https://nodejs.org/**](https://nodejs.org/)

##### https://mongoosejs.com/

* + [**https://www.npmjs.com/package/multer**](https://www.npmjs.com/package/multer)

##### https://developer.mozilla.org/

* + **https://getbootstrap.com/**
  + **http**[**s://www.w3schools.com/**](http://www.w3schools.com/)
  + **http**[**s://www.freecodecamp.org/**](http://www.freecodecamp.org/)
  + [**https://github.com/**](https://github.com/)