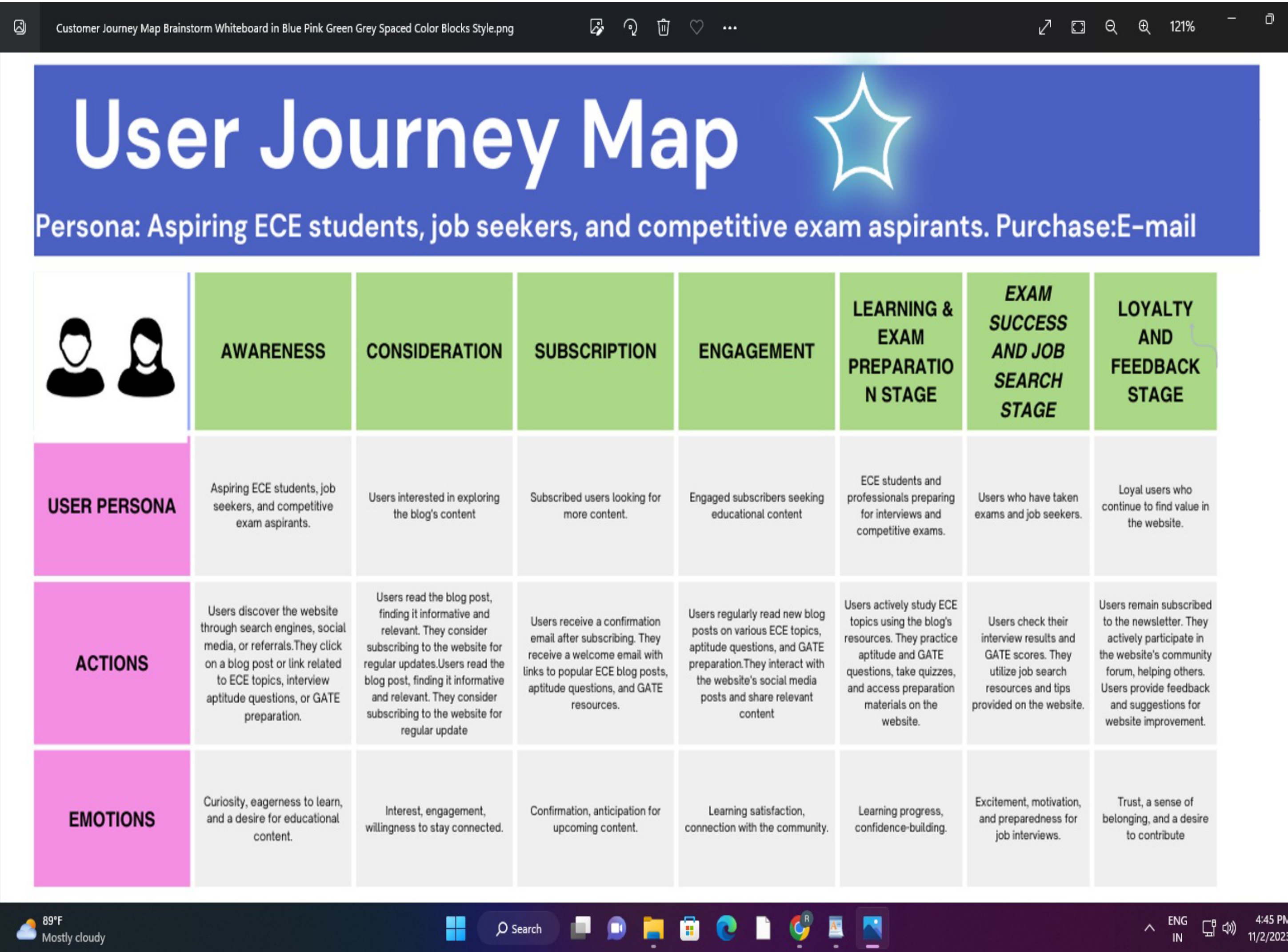


Project Design Phase-II

Solution Requirements (Functional & Non-functional)

Date	03 nov 2023
Team ID	NM2023TMID11237
Project Name	BUILDING A WEBSITE USING CANVA

CUSTOMER JOURNEY MAP:



Requirement analysis(functional,operational,technical)/ flowcharts

FUNTIONAL REQUIREMENTS:

1.User Registration and Authentication:

Users should be able to register accounts, log in, and manage their profiles.

2.Content Management:

Admins should be able to create, edit, and delete blog posts.

Authors should have the ability to create and edit their own posts.

Users should be able to comment on blog posts.

3.Content Categorization:

Posts should be categorized by topics, such as electron devices, digital electronics, microprocessors, microcontrollers, and aptitude questions.

4.Search Functionality:

Users should be able to search for specific blog posts and topics.

5.Social Media Integration:

Include social sharing buttons on blog posts for easy sharing on platforms like Twitter, Facebook, and LinkedIn.

6.Newsletter Subscription:

Users should be able to subscribe to a newsletter for updates.

7.User Feedback:

Implement a feedback form for users to provide comments or report issues.

Operational Requirements:

1.Content Publishing Workflow:

Define a content creation and approval process for authors and administrators.

2.Content Moderation:

Implement a moderation system to review user-generated comments and posts for inappropriate content.

3.User Support:

Provide a support system to address user queries and issues.

4.Backup and Recovery:

Regularly back up the website data to ensure data recovery in case of issues.

Technical Requirements:

1.Hosting and Domain:

Choose a reliable web hosting provider and register a domain name.

2.Content Management System (CMS):

Select a CMS like WordPress to facilitate content management.

3.Mobile Responsiveness:

Ensure that the website is responsive and accessible on various devices and screen sizes.

4.Database:

Use a database system to store user data, blog content, and comments.

5.Security:

Implement security measures to protect user data and prevent unauthorized access.

6.SEO Optimization:

Optimize the website for search engines with proper metadata and SEO best practices.

7.Performance:

Ensure fast loading times by optimizing images and scripts.

8.Analytics:

Integrate tools like Google Analytics to track website performance and user behavior.

TECHNICAL ARCHITECTURE

1. Front-End Architecture:

Web Development Framework: Choose a web development framework that aligns with your project requirements. Common choices include React, Angular, or Vue.js.

User Interface (UI): Develop responsive, user-friendly UI components for your website. Canva can be used to design custom graphics, banners, and featured images to enhance the visual appeal of your site.

User Experience (UX): Ensure a smooth and intuitive user experience, including easy navigation, accessibility, and mobile responsiveness.

Content Delivery: Use Content Delivery Networks (CDNs) to serve static assets, such as images and videos, efficiently to users around the world.

2. Back-End Architecture:

Server and Hosting: Choose a reliable hosting service that provides the necessary server resources to handle website traffic. Options include AWS, Google Cloud, or a managed WordPress hosting service.

Content Management System (CMS): Consider using a CMS like WordPress, which can simplify content creation and management.

Database: Set up a database system (e.g., MySQL, PostgreSQL) to store user data, blog posts, and comments.

Server-Side Framework: Use a server-side framework (e.g., Node.js, Ruby on Rails, Django) to handle server-side logic and API endpoints.

User Authentication: Implement user registration and authentication using technologies like OAuth, JWT, or a custom user management system.

Search Engine Optimization (SEO): Apply SEO best practices to improve search engine rankings and discoverability of your content.

3. Content Management:

Authoring and Editing: Use Canva for designing custom graphics and visuals to enhance your blog posts.

Workflow: Establish a content creation and approval workflow for authors and administrators.

Categorization: Create a taxonomy system to categorize and tag blog posts by topics, such as electron devices, digital electronics, microprocessors, microcontrollers, and aptitude questions.

Comments and User Interaction: Implement a system for users to comment on blog posts and interact with content.

4. Security:

SSL Certificate: Ensure secure data transmission by installing an SSL certificate for HTTPS.

Data Protection: Implement data security measures to protect user information and secure the database.

Firewall and DDoS Protection: Use firewalls and DDoS mitigation services to protect against cyber threats.

Regular Updates: Keep your CMS, plugins, and server software up to date to address security vulnerabilities.

5. Performance:

Caching: Implement caching mechanisms to improve website speed.

Image Optimization: Optimize images for web use to reduce page load times.

Content Delivery: Use CDNs to distribute content efficiently.

6. Analytics and Monitoring:

Analytics Tools: Integrate tools like Google Analytics to track website performance, user behavior, and traffic sources.

Error Logging: Implement error logging and monitoring to identify and address issues in real time.

7. Scalability:

Scalable Infrastructure: Plan for scalability as your website grows by choosing a hosting solution that can accommodate increased traffic and data.

Load Balancing: Use load balancing to distribute traffic evenly among multiple servers.

8. SEO Optimization:

SEO Best Practices: Implement SEO best practices, such as metadata optimization, schema markup, and XML sitemaps.

Keyword Research: Conduct keyword research to target relevant search terms in your content.

Open source frameworks

1.WordPress:

WordPress is a popular open-source content management system (CMS) that's highly customizable and user-friendly. It's well-suited for blogs and content-focused websites.

You can use Canva to create custom graphics and visuals for your WordPress posts and pages.

2.Jekyll:

Jekyll is a static site generator that is great for creating simple and fast websites. It's especially useful for blogs.

You can design your site's layout and visuals using Canva and then integrate them into your Jekyll templates.

3.Hugo:

Hugo is another static site generator that offers speed and simplicity. It's suitable for blogs and technical content websites.

You can design visuals with Canva and incorporate them into your Hugo themes.

4.Gatsby:

Gatsby is a static site generator that uses React to build blazing-fast websites. It's a good choice for creating content-rich websites with advanced features.

Canva designs can be integrated into your Gatsby website to enhance the visual appeal.

5.Ghost:

Ghost is a lightweight and open-source CMS designed specifically for blogs and publications. It's known for its simplicity and user-friendliness.

Canva can be used to create custom graphics for your Ghost blog.

6.strapi:

Strapi is a headless CMS that provides a flexible and customizable content management system. It's suitable for handling complex content structures.

You can design your website visuals with Canva and then integrate them into your Strapi-powered website.

7.Django:

Django is a high-level Python web framework that offers flexibility and scalability. It's a good choice for building complex websites with custom features.

You can use Canva for designing visual elements and then incorporate them into your Django templates.

8.Ruby on Rails:

Ruby on Rails is a web application framework that is known for its rapid development capabilities. It's suitable for building content-rich websites.

Canva designs can be integrated into your Ruby on Rails application for improved aesthetics.

Next.js (React Framework):

If you prefer a React-based solution, you can use Next.js to build a dynamic website with server-side rendering.

Canva visuals can be seamlessly incorporated into your Next.js project.

Third-party API'S

1.Social Media Sharing APIs:

APIs from social media platforms like Facebook, Twitter, and LinkedIn can be integrated to enable users to easily share your blog posts and content on their social profiles.

2.Commenting and Discussion APIs:

Services like Disqus and Facebook Comments offer APIs to add comment sections to your blog posts, allowing users to engage in discussions and provide feedback.

3.Email Marketing APIs:

Integration with email marketing services like MailChimp or SendGrid can help you build and manage your newsletter subscription list, sending out email updates to your subscribers.

4.Search APIs:

Implement a search functionality using search engine APIs like Algolia or Elasticsearch to help users find relevant content quickly.

5.Analytics and Tracking APIs:

Google Analytics or Matomo (formerly Piwik) can be integrated for tracking user behavior, site performance, and traffic analysis.

6.Content Delivery APIs:

Use a Content Delivery Network (CDN) like Cloudflare or Akamai to optimize content delivery for faster load times.

7.Payment and Donation APIs:

If you plan to offer premium content or accept donations, consider integrating payment gateways like PayPal or Stripe.

8.Authentication APIs:

Services like Auth0 or Firebase Authentication can provide secure user authentication and authorization for your website.

9.SEO APIs:

SEO tools like SEMrush or Moz offer APIs to help optimize your website's SEO and track keyword rankings.

10.Image and Media APIs:

Canva itself offers an API that you can use to create, edit, and manage custom graphics and images for your website directly from your Canva account.

11.RSS Feed APIs:

Implement RSS feeds that users can subscribe to in order to receive updates on new blog posts.

12.Calendar and Event APIs:

Integrate event calendars using services like Google Calendar or Eventbrite to announce and manage events, webinars, or workshops.

13.Translation APIs:

Offer content in multiple languages with translation APIs like Google Translate or Microsoft Translator.

14.Video Hosting APIs:

If you plan to include video content, integrate video hosting services like YouTube or Vimeo to embed videos in your blog posts.

15.Weather APIs:

Display relevant weather information if your website's content is location-specific or weather-dependent.

16.Code and Syntax Highlighting APIs:

Utilize code highlighting APIs like Prism or Highlight.js to improve the presentation of code snippets in technical blog posts.

17.Q&A and Forums APIs:

Services like Stack Exchange, Discourse, or Reddit offer APIs for creating Q&A sections or forums on your website.

18.Educational and Learning APIs:

Integrate APIs from e-learning platforms like Coursera or edX to offer courses and educational content related to electronics and communication engineering.

Select the APIs that align with your website's specific features and objectives. Remember to review each API's documentation and terms of use, and ensure that they provide the functionality you need while complying with data privacy and security requirements.

Cloud deployment:

1. Choose a Cloud Service Provider:

Select a cloud service provider that suits your requirements. Some popular options are:

Amazon Web Services (AWS)

Google Cloud Platform (GCP)

Microsoft Azure

Heroku

DigitalOcean

Consider factors such as pricing, availability in your region, and the specific services offered by each provider.

2. Register a Domain:

If you haven't already, register a domain name for your website. You can do this through a domain registrar or some cloud providers offer domain registration services.

3. Configure DNS Settings:

Set up your domain's DNS records to point to the cloud provider's resources. This typically involves configuring A or CNAME records to direct traffic to your website.

4. Create a Web Server or Hosting Environment:

Depending on the cloud provider you choose, set up a web server or hosting environment. This can involve the following steps:

Create a virtual machine (VM) or an instance (e.g., AWS EC2, GCP Compute Engine).

Choose an operating system (e.g., Linux, Windows) and configure it.

Install a web server (e.g., Apache, Nginx) or use a platform-as-a-service (PaaS) solution if available.

5. Install and Configure Your CMS:

Install and configure your chosen content management system (e.g., WordPress, Jekyll, Ghost) on your web server. Configure your CMS for your specific content and design requirements.

6. Secure Your Website:

Implement security best practices:

Enable SSL/TLS certificates to provide secure HTTPS connections.

Implement a Web Application Firewall (WAF) to protect against web threats.

Regularly update your CMS, plugins, and server software to patch security vulnerabilities.

7. Database Setup:

If your website requires a database, create and configure it. You can use managed database services offered by cloud providers to simplify this process.

8. Upload and Manage Content:

Migrate your existing content or start creating new content. You can use Canva to design visuals and graphics for your blog posts.

9. Performance Optimization:

Optimize your website's performance:

Use a Content Delivery Network (CDN) to distribute content globally.

Implement caching to reduce load times.

Optimize images for web use.

10. Backups and Disaster Recovery:

Set up regular backups of your website and database to ensure data recovery in case of issues. Cloud providers often offer backup and snapshot services.

11. Scalability:

Configure auto-scaling if your cloud provider supports it to handle traffic spikes. This ensures your website can scale up or down based on demand.

12. Monitoring and Analytics:

Use cloud monitoring and analytics services to track website performance, user behavior, and error logs. This data helps in making informed decisions.

13. Disaster Recovery Plan:

Develop a disaster recovery plan outlining steps to follow in case of a catastrophic event or data loss. Ensure backups are securely stored off-site.

14. Budget Management:

Keep an eye on your cloud usage and expenses. Set up cost management tools and alerts to prevent unexpected charges.

15. Testing:

Before going live, thoroughly test your website to ensure all functionalities are working as expected.

16. Go Live:

Once everything is set up, point your domain to your cloud-hosted website and make it live.

Cloud deployment offers the flexibility to scale resources as needed, ensuring that your website remains responsive and reliable, even during high traffic periods. Regularly maintain and update your website to provide a seamless experience for your visitors.