TESTIMONIAL–Web Application – Self project 1

Creating web application for testimonials that can be written by people I know. However, I would able to know through email whenever they send their comments about me. Once I give approval through mail, the automated message will be sent to the commenter and then in 2-3 days, these can be automatically printed in my portfolio website. Any idea how to implement it?

ChatGPT

There are several steps involved in implementing the web application you described. Here are the high-level steps and some suggestions on how to achieve them:

1. Design the application: You will need to design the user interface for your web application. This can be done using tools such as Figma, Sketch, or Adobe XD.
2. Develop the application: You can use web development frameworks such as React or Angular to develop your web application. You will also need to integrate a backend system for handling user authentication, email notifications, and storing the testimonials.
3. Add email notification functionality: You will need to use an email service such as SendGrid, Mailchimp, or Amazon SES to send email notifications whenever someone submits a testimonial. You can use their APIs to integrate email notification functionality into your application.
4. Approve and publish testimonials: Once you receive a testimonial, you will need to approve it before publishing it on your website. You can use a simple approval system where you receive an email notification with a link to approve the testimonial. Once approved, the testimonial can be automatically added to your portfolio website.
5. Automate printing: You can use a printing service API such as Printful or Printify to automatically print your testimonials on your website. You will need to create a custom script to retrieve the testimonials from your backend system and send them to the printing service API.

Overall, this project involves a fair amount of technical complexity, including web development, email service integration, and backend system setup. If you are not familiar with these technologies, you may want to consider hiring a developer or learning these skills yourself before starting the project.

What are the skills I need to know to implement the above webapp

ChatGPT

To implement the web application you described, you will need to have a good understanding of the following skills:

1. Front-end web development: You will need to know HTML, CSS, and JavaScript to build the user interface and functionality of your web application.
2. Web development frameworks: You can use web development frameworks such as React, Angular, or Vue.js to build your web application faster and more efficiently.
3. Backend web development: You will need to have a good understanding of backend web development to handle user authentication, database management, and server-side logic. You can use backend web development frameworks such as Node.js, Django, or Ruby on Rails to build the backend of your web application.
4. Database management: You will need to know how to set up and manage a database to store the testimonials and user information.
5. Email service integration: You will need to know how to integrate an email service such as SendGrid, Mailchimp, or Amazon SES to send email notifications when someone submits a testimonial.
6. API integration: You will need to know how to use APIs to integrate your web application with external services such as the printing service API.
7. Git and version control: You will need to know how to use Git and version control to manage your code and collaborate with other developers.
8. Basic design principles: You will need to have a basic understanding of design principles to create a user-friendly and visually appealing user interface.

These are some of the essential skills you will need to implement the web application you described. It may take some time to learn and master these skills, but with dedication and practice, you can create a high-quality web application.

2 / 2

Frame the roadmap with the duration in terms of days for the above skills to implement the application described above

ChatGPT

The duration required to learn and master the skills needed to implement the web application you described can vary depending on your previous experience and the amount of time you can dedicate to learning. However, here is a rough estimate of the time required to learn these skills:

1. Front-end web development (HTML, CSS, JavaScript): 20-30 days
2. Web development frameworks (React, Angular, Vue.js): 20-30 days
3. Backend web development (Node.js, Django, Ruby on Rails): 30-40 days
4. Database management: 20-30 days
5. Email service integration: 10-20 days
6. API integration: 10-20 days
7. Git and version control: 5-10 days
8. Basic design principles: 10-20 days

Note that these estimates are based on a rough approximation and can vary depending on your learning pace, the resources you use to learn, and the complexity of your web application. It is also important to note that these skills are interdependent and that you may need to learn them simultaneously or sequentially.

In summary, learning and mastering these skills can take approximately 125-200 days (about 4-7 months). However, it is essential to keep in mind that the learning process is ongoing, and you will continue to learn and improve your skills as you develop your web application and gain more experience.

Frame the roadmap for the above skills to implement the application described above

ChatGPT

2 / 2

Sure, here is a more detailed roadmap with steps and estimated timeframes to implement the web application you described:

1. Learn HTML, CSS, and JavaScript (20-30 days)
   * Learn the basics of HTML markup, CSS styling, and JavaScript programming
   * Practice building simple web pages and user interfaces
   * Familiarize yourself with front-end web development tools such as Visual Studio Code, Codepen, or Sublime Text
2. Choose a web development framework (2-3 days)
   * Research and compare different web development frameworks such as React, Angular, or Vue.js
   * Choose a framework that fits your project requirements and matches your skillset
   * Install and set up the chosen framework on your computer
3. Learn the chosen web development framework (20-30 days)
   * Learn the basics of the chosen framework, including its syntax and structure
   * Practice building simple applications and components using the framework
   * Explore the framework's documentation and community resources to deepen your understanding
4. Choose a backend web development framework (2-3 days)
   * Research and compare different backend web development frameworks such as Node.js, Django, or Ruby on Rails
   * Choose a framework that fits your project requirements and matches your skillset
   * Install and set up the chosen backend framework on your computer
5. Learn the chosen backend web development framework (30-40 days)
   * Learn the basics of the chosen backend framework, including its architecture and database management features
   * Practice building simple APIs and server-side logic using the framework
   * Explore the framework's documentation and community resources to deepen your understanding
6. Learn database management (20-30 days)
   * Learn the basics of database management, including data modelling, schema design, and CRUD operations
   * Practice building simple databases and queries using SQL or NoSQL databases
   * Explore popular database management tools such as MySQL, PostgreSQL, MongoDB, or Firebase
7. Learn email service integration (10-20 days)
   * Research and compare different email services such as SendGrid, Mailchimp, or Amazon SES
   * Learn how to integrate an email service into your web application using the service's API or SDK
   * Practice sending and receiving emails using the email service
8. Learn API integration (10-20 days)
   * Research and compare different APIs such as the printing service API or the authentication service API
   * Learn how to use APIs to integrate your web application with external services
   * Practice building simple API integrations using HTTP requests and response handling
9. Learn Git and version control (5-10 days)
   * Learn the basics of Git and version control, including creating and managing repositories, committing changes, branching, and merging
   * Practice using Git on your local computer and collaborating with other developers on GitHub or GitLab
10. Learn basic design principles (10-20 days)

* Learn the basics of design principles such as colour theory, typography, layout, and user experience
* Practice designing simple user interfaces and web pages using design tools such as Figma, Sketch, or Adobe XD

In summary, the roadmap to implement the web application you described can take approximately 145-243 days (about 5-8 months) depending on your learning pace, the resources you use to learn, and the complexity of your web application. Keep in mind that this is just an estimate, and the actual time required can vary. It is also important to note that learning these skills is an ongoing process, and you will continue to improve and refine them as you gain more experience and work on more projects.