# TỔNG QUAN

**1. Tổng quan tình hình nghiên cứu**

**2**. **Tổng quan về ReactJS.**

* ReactJS là một thư viện chứa nhiều JavaScript mã nguồn mở được Facebook xây dựng và phát triển. Thư viện này được sử dụng để tạo ra các ứng dụng trang web hấp dẫn với hiệu quả cao, tốc độ load nhanh và mã tối thiểu. Mỗi website sử dụng ReactJS phải chạy nhanh, mượt và có khả năng mở rộng cao, thao tác thực hiện đơn giản.
* Một số lợi ích khi sử dụng ReactJS:
  + ReactJS tạo ra cho bản thân nó một Dom ảo, nơi các Component được tồn tại trên đó và việc tạo ra Dom giúp cải thiện hiệu suất làm việc tốt hơn. Khi bạn cần cập nhật các thông tin lên Dom hoặc thay đổi gì đó, ReactJS đều có thể tính toán trước và thực hiện chúng. Nhờ đó mà ReactJS sẽ tránh được các thao tác cần có trên Dom và không tốn thêm bất cứ hành động nào khác.
  + Việc viết code trở nên dễ dàng hơn bởi nó sử dụng một cú pháp đặc biệt là JSX, cho phép trộn được giữa code HTML và JavaScript. Bên cạnh đó người dùng có thể sử dụng đoạn code này để thêm vào hàm Render mà không cần thực hiện việc nối chuỗi. Điều này được đánh giá là một trong những đặc tính mới cực kỳ thú vị của ReactJS. Đồng thời, việc chuyển đổi các đoạn HTML thành các hàm khởi động đều được thực hiện từ bộ biến đổi chính là JSX.
* Ưu điểm của ReactJS:
  + Hiệu suất cao: ReactJS sử dụng Virtual DOM để tối ưu hóa hiệu suất, giúp cập nhật giao diện người dùng một cách hiệu quả hơn và giảm tải cho DOM thực sự.
  + ReactJS cho phép bạn phân chia giao diện thành các thành phần độc lập, có thể tái sử dụng dễ dàng, giúp giảm thiểu mã lặp lại và tăng khả năng bảo trì của mã.
  + ReactJS có một cộng đồng lớn, nhiệt tình và chia sẻ kiến thức rộng rãi, cung cấp hỗ trợ và tài liệu phong phú cho nhà phát triển.
* Nhược điểm của ReactJS:
  + Reactjs không phải là framework, cho nên phải tự xây dựng dự án bằng thủ công.
  + Tích hợp Reactjs vào các framework MVC truyền thống yêu cầu cần phải cấu hình lại
  + Poor Document: Đó là một nhược điểm khá phổ biến đối với các công nghệ cập nhật liên tục. Các công nghệ cập nhật và tăng tốc nhanh đến mức không có thời gian để tạo tài liệu phù hợp. Để khắc phục điều này, các nhà phát triển tự viết hướng dẫn với việc phát triển các bản phát hành và công cụ mới trong các dự án hiện tại của họ.

**3. Tổng quan về Django REST Framework**

* Django REST Framework (DRF) là một framework mạnh mẽ được xây dựng trên nền tảng của Django, được sử dụng để phát triển các API RESTful trong các ứng dụng web Python.
* Ưu điểm của Django REST Framework:
  + Dễ sử dụng: DRF cung cấp các công cụ và thư viện mạnh mẽ để phát triển các API RESTful một cách nhanh chóng và dễ dàng. Nó tích hợp chặt chẽ với Django, giúp giảm thiểu thời gian phát triển.
  + Hỗ trợ tài liệu tốt: DRF đi kèm với một bộ công cụ tài liệu mạnh mẽ, bao gồm trang tài liệu tự động được tạo ra từ các serializers, viewsets và các chế độ xem API khác. Điều này giúp cung cấp tài liệu rõ ràng và dễ hiểu cho các API mà bạn tạo ra.
  + Tương thích tốt với Django: DRF được thiết kế để hoạt động chặt chẽ với Django, tận dụng các tính năng và cấu trúc của Django như models, views và templates.
* Nhược điểm của Django REST Framework:
  + Cấu hình phức tạp: Mặc dù DRF cung cấp nhiều tính năng mạnh mẽ, nhưng việc cấu hình và tuỳ chỉnh có thể phức tạp đối với những dự án lớn và phức tạp.

Khó khăn trong việc quản lý hiệu suất: Trong một số trường hợp, việc quản lý hiệu suất của các API DRF có thể trở nên khó khăn khi xử lý lượng truy cập lớn hoặc yêu cầu tính toán phức tạp.