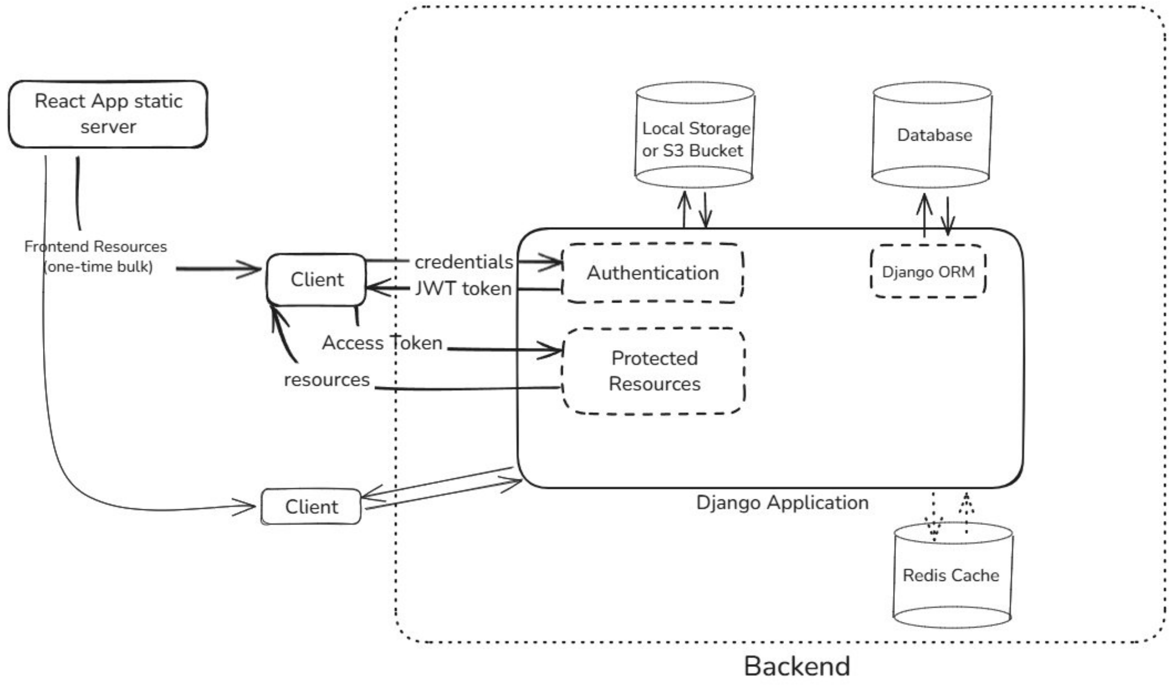
Tech Stack and Architecture Choice:



ARCHITECTURAL DIAGRAM

- Frameworks used – Django Rest Framework for backend and React.JS for frontend.
  - React is a go-to choice for SPA.
  - Django has good in-built support for authentication, caching and ORM to handle SQL data
  - SQLite DB is used for a light weight DB server
- Data Flow
  - Register a new account and get your username.
  - Login with username and password
  - Choose your role – Patient or Clinician
  - Patient uploads a pdf file and additional information related to the file. The file is saved to local storage on the server side. The metadata is stored into file-metadata table along with unique id and file path.
  - When a patient or clinician chooses to download a file, a call is made to the file-metadata table with file id parameter and the file is sent in the response.
  - Patient can delete the uploaded documents

- API endpoints
  - POST /login
  - POST /register
  - POST /register-patient
  - POST /register-clinician
  - GET /users/:username
  - GET /patients
  - GET /patients/:id
  - GET /clinicians
  - GET /clinicians/:id
  - POST /patients/:id/documents/upload
  - GET /patients/:id/documents/:id
  - DELETE /patients/:id/documents/:id
  - GET /patients/:id/documents
  - GET /patients/:id/documents/:id/download
- Key Considerations
  - Scalable – Open to scale
  - Files are stored locally. Can be extended to use S3 bucket.
  - Security – Authorization and Permissions are validated before sending the API reponse.