**Web App to register persons with disabilities**

* **Objective**
* **Technologies used**
* **Overall process**

1. **Objective**

**The objective of the project is to make a app that helps the us to Register , Login and view the patients or persons with disabilities.**

**We register a person Name ,Phone ,Email ,Address and his/her disabilities**

**The app help the user to login and then check his/her information.**

1. **Technologies Used**

**Python**

**Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.**

**Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.**

**Everyday tasks**

**Python isn't only for programmers and data scientists. Learning Python can open new possibilities for those in less data-heavy professions, like journalists, small business owners, or social media marketers. Python can also enable non-programmer to simplify certain tasks in their lives. Here are just a few of the tasks you could automate with Python:**

**Keep track of stock market or crypto prices**

**Send yourself a text reminder to carry an umbrella anytime it’s raining**

**Update your grocery shopping list**

**Renaming large batches of files**

**Converting text files to spreadsheets**

**Randomly assign chores to family members**

**Fill out online forms automatically**

**Why Python?**

**Python is popular for a number of reasons. Here’s a deeper look at what makes it so versatile and easy to use for coders.**

**It has a simple syntax that mimics natural language, so it’s easier to read and understand. This makes it quicker to build projects, and faster to improve on them.**

**It’s versatile. Python can be used for many different tasks, from web development to machine learning.**

**It’s beginner friendly, making it popular for entry-level coders.**

**It’s open source, which means it’s free to use and distribute, even for commercial purposes.**

**Python’s archive of modules and libraries—bundles of code that third-party users have created to expand Python’s capabilities—is vast and growing.**

**Python has a large and active community that contributes to Python’s pool of modules and libraries, and acts as a helpful resource for other programmers. The vast support community means that if coders run into a stumbling block, finding a solution is relatively easy; somebody is bound to have run into the same problem before.**

**Ready to get started? Learn the basics of Python with the Python for Everybody Specialization from the University of Michigan—no previous coding experience is needed.**

**Tornado**

**Tornado is a Python web framework and asynchronous networking library, originally developed at FriendFeed. By using non-blocking network I/O, Tornado can scale to tens of thousands of open connections, making it ideal for long polling, WebSockets, and other applications that require a long-lived connection to each user.**

**Tornado is different from most Python web frameworks. It is not based on WSGI, and it is typically run with only one thread per process. See the User’s guide for more on Tornado’s approach to asynchronous programming.**

**While some support of WSGI is available in the tornado.wsgi module, it is not a focus of development and most applications should be written to use Tornado’s own interfaces (such as tornado.web) directly instead of using WSGI.**

**In general, Tornado code is not thread-safe. The only method in Tornado that is safe to call from other threads is IOLoop.add\_callback. You can also use IOLoop.run\_in\_executor to asynchronously run a blocking function on another thread, but note that the function passed to run\_in\_executor should avoid referencing any Tornado objects. run\_in\_executor is the recommended way to interact with blocking code.**

**Why use Tornado?**

**Tornado doesn’t get in your way**

**Tornado provides a set of libraries that make it easy to do the hard parts of building a web application. Since tornado provides only the libraries you need, your code remains clean. You don’t have to fight with a large framework as you would with other web development frameworks.**

**Ease to scale**

**By using non-blocking network I/O, Tornado can scale to tens of thousands of open connections, making it ideal for long polling, WebSockets, and other applications that require a long-lived connection to each user.**

**Tornado is one of the most prominent libraries in Python for developers who build high-performance, low latency web applications.**

**If you are a Python developer and you haven’t heard of Tornado, then you are probably living under a rock (don’t worry, it’s cool). And there are more reasons why you should consider using Tornado instead of other frameworks, especially if your application has high demands on performance and scalability.**

**Cryptography**

**cryptography is a package which provides cryptographic recipes and primitives to Python developers. Our goal is for it to be your “cryptographic standard library”. It supports Python 3.6+ and PyPy3 7.2+.**

**cryptography includes both high level recipes and low level interfaces to common cryptographic algorithms such as symmetric ciphers, message digests, and key derivation functions. For example, to encrypt something with cryptography’s high level symmetric encryption recipe:**

**Why Cryptography?**

**Cryptography is an essential information security tool. It provides the four most basic services of information security −**

**Confidentiality − Encryption technique can guard the information and communication from unauthorized revelation and access of information.**

**Authentication − The cryptographic techniques such as MAC and digital signatures can protect information against spoofing and forgeries.**

**Data Integrity − The cryptographic hash functions are playing vital role in assuring the users about the data integrity.**

**Non-repudiation − The digital signature provides the non-repudiation service to guard against the dispute that may arise due to denial of passing message by the sender.**

**All these fundamental services offered by cryptography has enabled the conduct of business over the networks using the computer systems in extremely efficient and effective manner.**

**MogoDb**

**MongoDB is a document database used to build highly available and scalable internet applications. With its flexible schema approach, it’s popular with development teams using agile methodologies. Offering drivers for all major programming languages, MongoDB allows you to immediately start building your application without spending time configuring a database.**

**MongoDB is an open-source document database built on a horizontal scale-out architecture that uses a flexible schema for storing data. Founded in 2007, MongoDB has a worldwide following in the developer community.**

**Instead of storing data in tables of rows or columns like SQL databases, each record in a MongoDB database is a document described in BSON, a binary representation of the data. Applications can then retrieve this information in a JSON format.**

**Why Use MongoDB?**

**MongoDB is built on a scale-out architecture that has become popular with developers of all kinds for developing scalable applications with evolving data schemas.**

**As a document database, MongoDB makes it easy for developers to store structured or unstructured data. It uses a JSON-like format to store documents. This format directly maps to native objects in most modern programming languages, making it a natural choice for developers, as they don’t need to think about normalizing data. MongoDB can also handle high volume and can scale both vertically or horizontally to accommodate large data loads.**

**MongoDB was built for people building internet and business applications who need to evolve quickly and scale elegantly. Companies and development teams of all sizes use MongoDB for a wide variety of reasons.**

1. **Overall Process**

**Using tornado framework we have set up an app.  
With tornado templating we show HTML pages to user to Register and Login into the tornado app. Using different handlers in the tornado view file we contral what to show and url file getting all the requests from the app and sends it to view file.View file renders html pages and handles the seesion of the user.**

**By sending request into the app.A user can login anto the account.**

**The session of the logged in user is controlled by tornado in built session functions,**

**The data of the users in stored using Mongo Database and connectivity between mongo and tornado is by using pymongo library.  
the password of users are encrypted using python cryptography library.**